Computers and Information Technology
2024

MOODUP – A SMART AMBIENT LIGHTING SYSTEM USING SENTIMENT ANALYSIS CONTROLLED BY A MOBILE CHATTING APPLICATION

Candidate: Sorina-Andreea POPA

Supervisor: conf.dr.ing. Răzvan BOGDAN

Session: June 2024

REZUMAT

În această eră modernă, stresul și anxietatea devin probleme majore în mediul educațional, profesional și de zi cu zi. Psihologii au demonstrat că lumina are un efect pozitiv asupra stării de spirit și poate reduce aceste simptome. Alți factor care îmbunătățesc starea de bine sunt temperatura, umiditatea si lumina naturală a unei camere. Această lucrare de licență prezintă proiectarea și dezvoltarea unui Sistem Inteligent de Iluminare Ambientală care îmbunătățește bunăstarea utilizatorului prin ajustarea luminii în funcție de stările emoționale ale acestuia și prin monitorizarea încăperii. Prin integrarea unei aplicații mobile ce are la bază o conversație cu o Inteligență Artificială, analiza sentimentelor și componente hardware de monitorizare, sistemul oferă un mediu receptiv și de suport pentru utilizatori.

Aplicația bazată pe Android permite utilizatorilor să comunice cu un asistent cu Inteligență Artificială, iar starea lor emoțională este analizată folosind un algoritm de analiză a sentimentelor. Sistemul ajustează apoi lumina ambientală pentru a îmbunătăți starea de spirit a utilizatorului. Configurația hardware include senzori pentru monitorizarea temperaturii, umidității, intensității luminii și mișcării, asigurând condiții de mediu optime. Serviciile cloud, inclusiv Node.js și Firebase, facilitează transmisia și stocarea fără întrerupere a datelor. Testarea complete asigură siguranța și eficiența sistemului. Acest proiect își propune să sprijine persoanele care se confruntă cu probleme de sănătate mintală, oferind o soluție avansată tehnologic care combină inteligența emoțională cu controlul mediului.

ABSTRACT

In this modern age, stress and anxiety are becoming a huge issue in the educational, work, and everyday environment. It has been proven by psychologists that light has a positive effect on mood and can reduce these symptoms in people. Other environmental factors that improve the well-being are the temperature, humidity and natural lighting of a room. This thesis presents the design and development of a Smart Ambient Lighting System that enhances user well-being by adjusting lighting based on emotional state and monitoring the environment. By integrating a mobile application with conversational AI, sentiment analysis, and monitoring hardware components, the system provides a responsive and supportive environment for users.

The Android-based application allows users to communicate with an AI assistant, and their emotional state is analysed using a sentiment analysis algorithm. The system then adjusts the ambient lighting to improve the user's mood. The hardware setup includes sensors for monitoring temperature, humidity, light intensity, and motion, ensuring optimal environmental conditions. Cloud services, including Node.js and Firebase, facilitate seamless data transmission and storage. Comprehensive testing ensures the reliability and efficiency of the system. This project aims to support individuals dealing with mental health challenges by providing a technologically advanced solution that combines emotional intelligence with environmental control.

CONTENTS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
IoT	Internet of Things
IDE	Integrated Development Environment
UML	Unified Modeling Language

LIST OF FIGURES

LIST OF TABLES

LIST OF SNIPPETS

1. INTRODUCTION

1.1 CONTEXT

From the academic environment to job-related tasks, mood plays an immense role in people’s work performance in their activities, so it needs to be maintained in a positive state as long as possible [1].

One of the main factors that influences mood is the environment in which individuals work. Studies have shown that people are more efficient in favourable settings, with aspects like having a satisfying room temperature or getting enough natural light [2]. At the same time, from the paint on the walls to using LED light strips, the predominant colour of the setting has a profound impact on well-being. For instance, highly saturated or vibrant colours can be overwhelming and may induce feelings of stress and agitation [3].

Unfortunately, stress and anxiety are becoming increasingly prominent in this constantly changing modern age, a significant impact played by the COVID-19 pandemic. These issues can affect all ages, from children struggling with school, to young adults balancing academic and professional lives, to older adults managing family responsibilities [4]. And the previously mentioned issues are just a few examples of mental health challenges people struggle with today.

However, the advantage of the rapid advancements in technology offers the necessary tools to diminish these issues, the latest addition being the recent integration of OpenAI [5], which provides human-like interaction capabilities.

Keeping that in mind, this thesis proposes A Smart Ambient Lighting System that changes based on the mood of the user, giving them complete control over the setting they are working in, being able to monitor and change their environment. After creating an account and logging into the mobile app, whenever the user feels sad or stressed, they can communicate their struggles to an AI through a chat feature and receive soothing responses. Using a sentiment analysis algorithm based on the AFINN list of words [6], the system adjusts the room lighting based on the user’s mood, to improve their emotional state.

Therefore, this project targets individuals that struggle with mental health issues who might find it difficult to communicate their problems with others. The system provides a 24/7 companion, potentially enhancing their performance by creating a supportive environment.

1.2 MOTIVATION

In recent years, there has been a spike in mental health issues among children and teens, the COVID-19 pandemic being a considerable influence [7]. From needing to drastically change their lifestyle caused by the tragic incidents happening around them, to missing crucial developing moments of their lives – like going to high school or even graduation – the isolation happened not only physically, but mentally as well [8].

Moreover, teens tend to struggle with psychological issues, a dramatic increase in the past 20 to 30 years, influenced by social changes, like the rise in youth unemployment or pressure in school or in the workplace. Some significant examples of mental issues that they tend to struggle with are stress, anxiety, loneliness, depression, etc. Disruptive behaviours are more common in boys, whereas anxiety and depression are more common in girls [9].

These patterns have been observed personally, especially in the pandemic context. Anxiety had been in control in moments of online meetings, turning on the microphone to talk being one of the most demanding tasks. The stress caused by starting a conversation with someone new and the fear of not being able to make real connections with people over the internet were constantly present, these feelings leading to an undiagnosed state of fatigue and depression.

Evidently, mental health issues don't target only teens. Specifically for Romania, based on a study realised by REGINA MARIA [10], 55% of people say that they have felt a negative impact in their well-being caused by the recent events happening around the world, but as shown in Figure 1.1, 89% of individuals have never sought any psychotherapy services. A few of the reasons why people refuse to go to therapy are the high prices, the lack of time, or the fact that they don't believe a stranger can solve their issues [11].

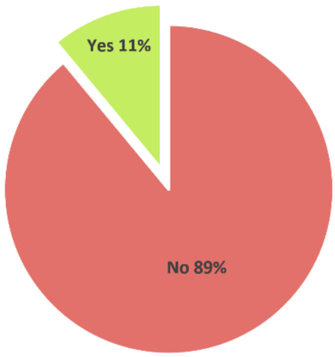

Figure 1.1: Romanian People Seeking Therapy

Taking advantage of the latest inclusion of the conversational AI tools, two popular examples being OpenAI [5] and Character.AI [12], there has been a significant impact of having such means in people's day-to-day life. Having constant reassurance when it is needed, a companion on standby always ready to give good feedback, people that are exposed to it show considerable improvement. After a trial with a specific mobile conversational AI tool, results have shown good improvement in stress and anxiety levels for the people who had been using it [13].

Considering these factors, there was a personal interest in finding a solution that would help the people that can't get officially diagnosed but would like to improve their mental state and productivity by having a place where they could lie their struggles and receive active feedback. Therefore, being a tool that could work in collaboration with certified psychotherapists, this thesis project offers a way to improve the well-being of a patient and a place to track their state of mind over multiple days.

1.3 THEORETICAL BACKGROUND

For satisfying results regarding work performance, it is important to take in consideration how environmental factors affect mood and which of these conditions to improve or change so that it enhances the state of mind. This thesis project focuses specifically on temperature, humidity, room lighting, and colour.

Productivity is highly influenced by the **temperature** of the working environment, being at its peak when the optimal thermal conditions are met. As well, temperature plays a role in influencing stress and energy levels. Optimal values may vary based on the mental health issues an individual struggles with. A spike in temperature may be advantageous for most people but might have opposite effects for certain individuals [14].

Humidity impacts the comfort considerably, especially in hot-humid conditions. Individuals typically prefer a more humid atmosphere over a dry one, so they tend to choose cooler indoor temperatures. The overall optimal humidity is about 50% [15].

Adequate **lighting** in the workplace can increase the comfort and visibility of individuals, which can boost overall performance. It helps reduce eye strain and fatigue, leading to less errors in tasks that require a high level of attention. Whereas if the environment light is too dim or too bright, the results have an opposite effect. Therefore, maintaining a proper illumination level is essential, the ideal illuminance level revolving around 146lux [16].

In Table 1 there can be observed a compilation of optimal values for temperature, humidity, and lighting in the work environment for increasing the work performance, based on the studies previously mentioned.

Table 1: Optimal Values for Increasing Work Performance

factor	optimal lower bound	optimal value	optimal upper bound
temperature	20°C	22°C	25°C
humidity	40%	53%	60%
lighting	100 lux	146lux	180 lux

In close association with light, more **colour** in the work environment displays favourable outcomes in people's performance, depending on lightness, vibrancy and hue. An effective colour scheme can boost an individual's mood, as Figure 1.2 shows, measuring the performance in different scenarios over multiple months [17].

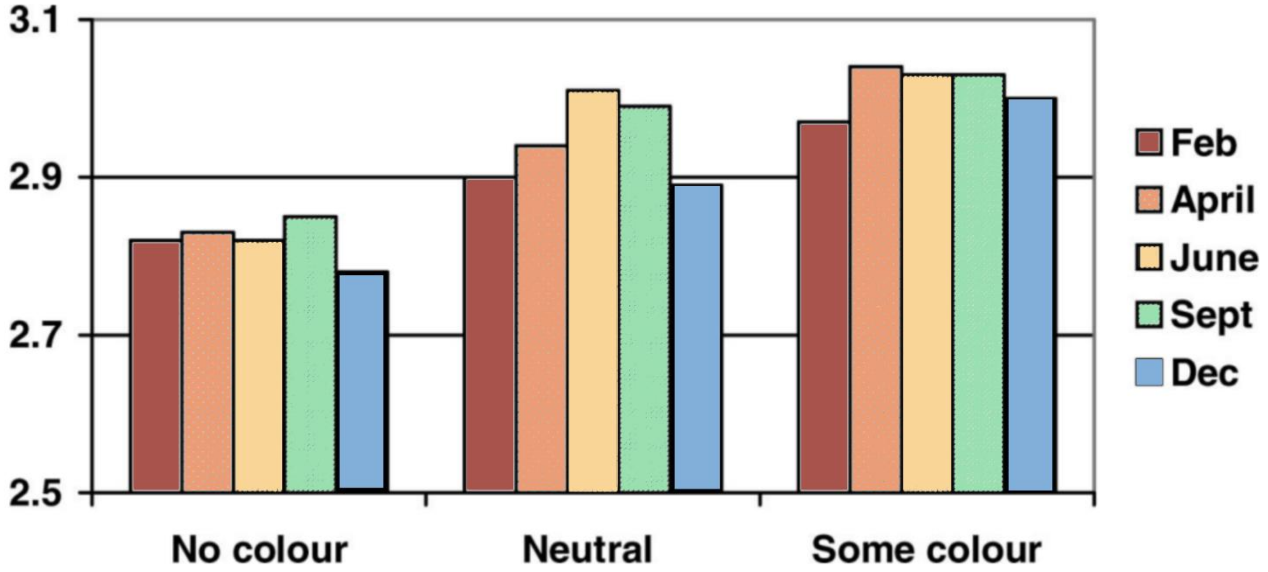

Figure 1.2: Mood Improvement for Different Levels of Colour in Work Environments [17]

Thus, to improve the mood of the user, it is important to know how emotions are associated with colour hues. Positive emotions are usually represented by light colours, the most popular being yellow to signify joy. When the chroma lowers, the association goes towards negative emotions. In Table 2 can be observed the breakdown for each colour and its associated emotion [18].

Table 2: Colour and Its Associated Emotion

colour	associated emotion
red	energy, agitation, arousal, confidence
orange	warmth, positive, joy, enthusiasm
yellow	joy, happiness, cheerfulness, delight
green	relaxation, calmness
cyan	calmness, peace, safety
blue	tranquillity, soothing
purple	joy, excitement
pink	love, tenderness
grey	fear, sadness
black	depression, sadness

This thesis project uses a spectrum of colours starting from hues of blue to induce calmness, offering a soothing feeling for a lower mood. As the well-being increases, the colours switch towards green hues and yellow for inducing relaxation and prolonging the state of joy, changing to shades of magenta as it reaches the maximum level of a good emotional state.

Thus, inadequate environmental factors will most likely influence people’s emotion in a negative way, worsening for those who struggle with mental illnesses. To prevent this, it is important for ambient parameters to be tracked and controlled according to the user’s preference, resulting in better performance and improving their living conditions [19], which is the exact target of this thesis project.

2. CURRENT STATUS IN THE FIELD

2.1 SCIENTIFIC PAPERS

There is a significant debate about whether conversational AI can deliver therapy services with the same effectiveness as a human therapist would. In order to do that, it is necessary to know what makes human-delivered psychotherapy effective being delivered by a non-human agent. While it is uncertain if AI can currently meet this expectation, the field is still under development and might get to that end goal in the coming years [20].

Currently, AI is being integrated into mobile applications in a chatbot form to enhance the experience of an individual who wants to undergo treatment, optimising personalised health care through natural language processing. This technology allows AI to understand human language based on sentiment and voice tone. There is significant growing evidence supporting the efficacy of these tools, with data indicating a reduction in symptoms of depression, anxiety and suicidal thoughts among users [21].

However, these apps often fail to meet the needs and expectations of the users, leading to low engagement. Users might feel sceptical about the effectiveness of AI interactions compared to human-to-human conversations. At the same time, there might be concerns about the safety and data handling of such sensitive information, as many tools collect user messages to further train the AI [22].

Therefore, to enhance the adoption and effectiveness of digital mental health tools, there are several approaches to be taken into consideration. One promising concept is the idea of a ‘digital clinic’, which combines the in-app conversational style with access to a real therapist. With the consent of the user, this way the AI tools can be trained with accurate professional data, bringing them closer to replicating human therapy interactions [23].

2.2 EXISTING TECHNOLOGIES

Taking in consideration popular operating systems, Android is currently at the top with over 70% running devices, being the best fit for this thesis project. Being an open-source operating system provided by Google, it is adapted for various devices, from smartphones to tablets and smart TVs. This extensive user base and flexibility makes Android an ideal choice for developing an application intended for a broad audience [24].

As this thesis project has multiple functionalities, there are similar technologies that can serve either as conversational AI applications, mental health applications, or smart home devices. The following analysis will strictly consider smart lighting system mobile apps currently present on the Romanian market.

Philips Hue [25] is one of the more popular and more expensive smart lighting systems that can change the colour and intensity of lights in the user's home based on their preference. Controlled through a mobile application, compatible with Google Home [26] and Alexa [27], accepting voice commands, the end goal is creating a customised lighting environment for different activities.

Ledvance Smart+ [28] is similar to the previously mentioned system, offering a more accessible price for their products. It has smart routines presets for different activities, and a function that is reactive to sound and music, changing based on the beat.

Using a **Tp-Link Tapo** [29] smart lightbulb personally, the application offers a range of product compatibility, allowing routine presets based on observational data gathered by video cameras.

For an individual to use the application, they first need to possess one of the products, in this example, a smart light bulb. After installing the app, the user is welcomed by an intuitive tutorial on how to connect the product to wi-fi. As it is shown in Figure 2.1 and Figure 2.2, after the setup is complete, the user can control the lighting of the room and set preferred routines. These routines might depend on owning other devices from the provider. As well, the user can see the runtime of the device, being able to monitor its energy consumption.

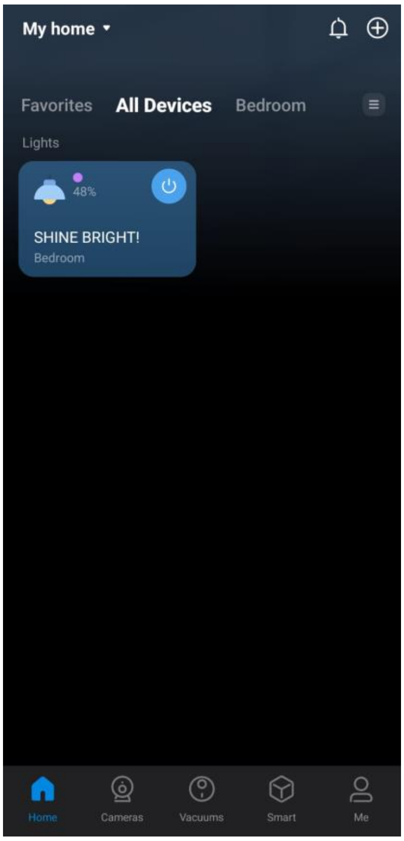

Figure 2.1: The Home Page of the Tapo Mobile App

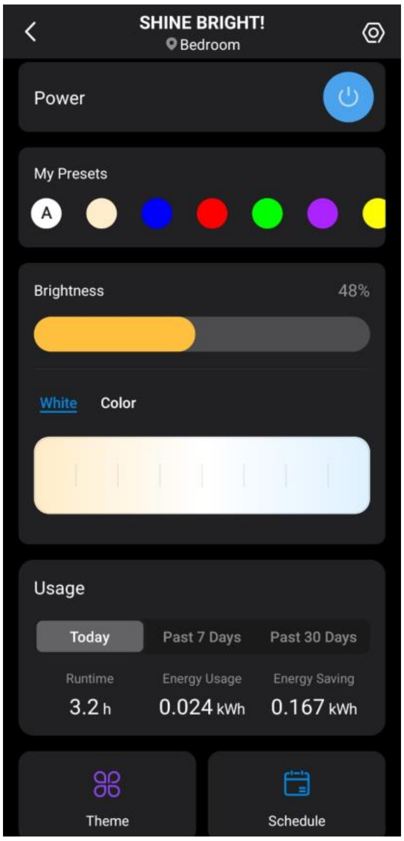

Figure 2.2: The Device Control Page of the Tapo Mobile App

2.3 WHAT THIS PROJECT BRINGS

Table 3: Comparison with Other Products on the Market

Feature/Application	Thesis project	Philips Hue	Ledvance Smart+	Tp-Link Tapo
Conversational AI	**Yes**	No	No	No
Sentiment Analysis	**Yes**	No	No	No
Automatic Mood-Based Lighting	**Yes**	No	No	No
Weekly Mood Analysis	**Yes**	No	No	No
Environmental Control	**Yes**	**Yes** (Requires additional device)	No	**Yes** (Requires additional device)
Temperature Monitoring	**Yes**	No	No	No
Humidity Monitoring	**Yes**	No	No	No
Natural Light Monitoring	**Yes**	No	No	No
Motion Based Switch	**Yes**	**Yes** (some products)	**Yes** (some products)	**Yes** (some products)
Routines Setup	No	Yes	Yes	Yes
Environmental Tips for Mood Enhancement	**Yes**	No	No	No
Cross Platform Compatibility	No	**Yes**	**Yes**	**Yes**
Home Assistant Compatibility	No	**Yes**	**Yes**	**Yes** (limited)

The proposed application combines the functionalities of conversational AI and smart home devices, integrating sentiment analysis to create a more empathetic and responsive user experience. Compared to the previously mentioned ideas, and technologies that focus solely on smart home control, MoodUp leverages the power of AI to understand and respond to the user's emotional state. This thesis project relies on the following key features:

1. **24/7 Companion**: the AI assistant provides permanent assistance, making users feel seen and understood, which is beneficial for individuals dealing with mental health issues, feedback being crucial.
2. **Sentiment Analysis**: the app adjusts the ambient lighting to reflect and improve the mood of the user by analysing their messages, creating a more supportive and personalised environment.
3. **Environment Control**: users can monitor and adjust their room's temperature, humidity, and lighting, ensuring optimal conditions for their well-being and productivity.

3. USED TECHNOLOGIES

3.1 SOFTWARE – MOBILE APP

3.1.1 ANDROID STUDIO

The chosen IDE for developing the mobile app is Android Studio [30]. Google provides all the necessary resources for learning Android app development through their platform, from online courses to YouTube tutorials, making it accessible for developers of all skill levels.

Having an intuitive interface and good live editing tools for simulating the application through an emulator or a physical device, Android Studio offers total control over the aspect and functionality of the app. Its integration with Firebase simplifies backend operations, enhancing app functionality and performance.

3.1.2 KOTLIN

The preferred programming language recommended by Google for Android Development is Kotlin [31], so this is the language the app is developed in. Having Java at its foundation, Kotlin offers easier-to-understand syntax, making the flow of reading and writing code easier. This improves developer productivity by reducing the number of errors, and Kotlin's interoperability with Java allows an easy switch from existing Java libraries, ensuring a smooth development process.

3.1.3 MVVM – MODEL-VIEW-VIEWMODEL ARCHITECTURE

The Model-View-ViewModel (MVVM) architecture [32] was chosen to structure the mobile application, because it encourages a clean way of separating the responsibilities inside the app, making the code files more manageable.

The model handles data, the View manages the visual part of the interface, and the ViewModel acts as a middle ground between the previous two. This type of architecture simplifies the process of linking the visual elements with the application logic, being suitable for the maintainability of the thesis app.

3.1.4 OpenAI API

Integrated through the Android Jetpack Compose, the OpenAI [5] GPT-3.5-turbo [33] API is utilised to generate the responses for the user's messages within the app. This API enables sophisticated natural language processing capabilities, allowing the app to provide interactive and engaging conversational experiences, answering in a human-like manner. This way, the user's messages can vary from asking simple questions to more emotionally based interactions.

3.2 HARDWARE – DEVICE

3.2.1 ESP32 MICROCONTROLLER

Offering Wi-Fi and Bluetooth connectivity, allowing direct communication to the database, the ESP32 microcontroller [34] is a well-fit central processing unit for this project. Its compact design and superior data transmission capabilities make it a more efficient alternative to an Arduino board. Its instructions are provided through the Arduino IDE with the help of the C++ programming language. It operates at 3.3V and includes a built-in voltage regulator, allowing it to be powered by either 5V or 3.3V.

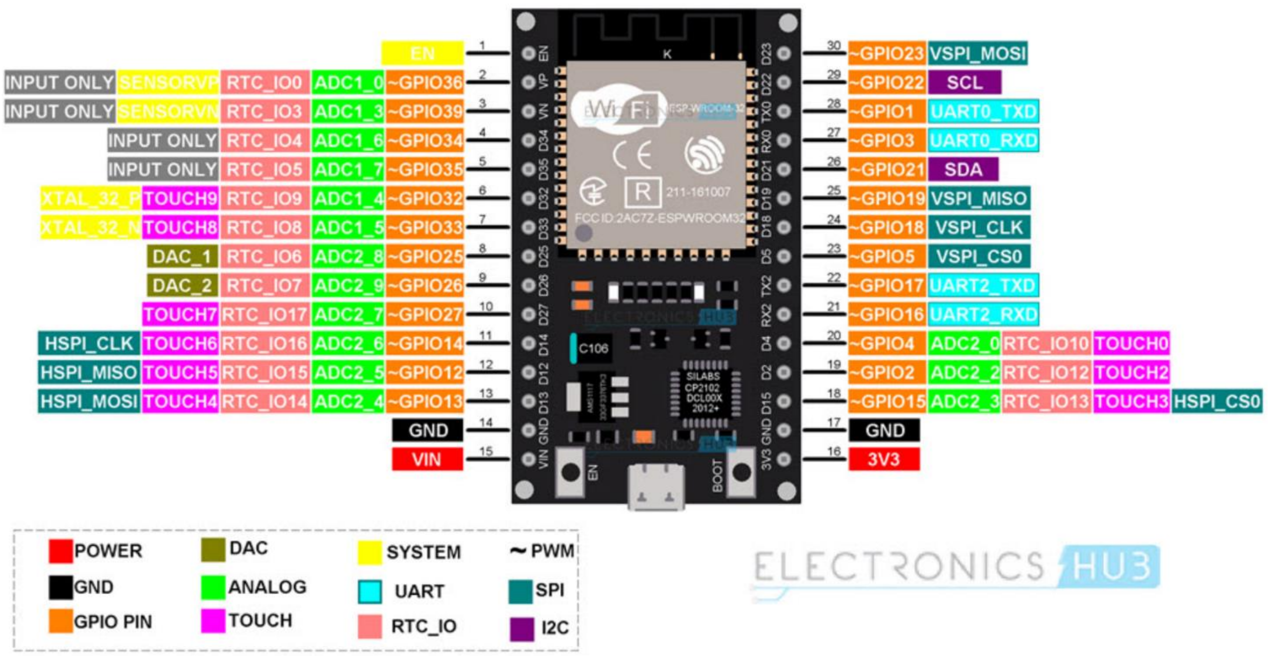

Figure 3.1: ESP32 Microcontroller IC [34]

3.2.2 DHT11 SENSOR MODULE

The DHT [35] is a digital temperature/humidity sensor module used for monitoring the environmental conditions of the room. Characteristics:

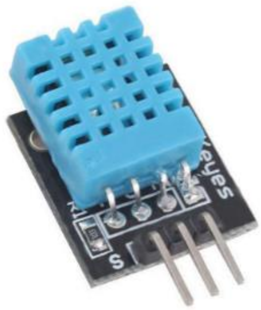

Figure 3.2: DHT11 Sensor Module [35]

- It has 3 pins for power, ground, and data output;
- It operates at voltages between 3.3V-5V;
- It measures temperatures between 0°C-50°C, ±1°C accuracy;
- It measures humidity between 20%-90%, ±1% accuracy;
- It doesn't record negative data;
- Output: Serial data.

3.2.3 PIR SENSOR MODULE

The HC-SR501 PIR (Passive Infrared) [36] is an analogue sensor module designed to detect the motion within the room. It works as a switch which activates the or deactivates the lighting system based on the presence of a person in a room. Characteristics:

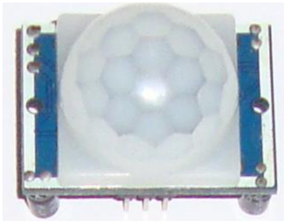

Figure 3.3: PIR Sensor Module [36]

- It has 3 pins for power, ground and data output;
- It operates at voltages between 3.3V-12V;
- It can make the difference between human and object movement;
- It can cover distances up to 7 metres to a 120° angle;
- It can operate on temperatures from -20°C to +80°C;
- Output: voltage high/low.

3.2.4 LIGHT SENSOR

The GY-302 BH1750 Ambient Light Sensor [37] is a digital light intensity sensor that measures the ambient light level in the room. This sensor is used to ensure that the room gets enough natural light. Characteristics:

Figure 3.4 - Light Sensor Module [37]

- It has 5 pins but only 4 of them are used: power, ground, serial clock and serial data;
- It operates at voltages between 3V-5V;
- It can measure light between 1-65535 lx, ±20% accuracy;
- It has a built-in analogue to digital converter (from analogue illuminance to digital data).

3.2.5 RGB MODULE

The RGB LED [38] module is the primary light source that changes colours based on the user's sentiment input calculated by the sentiment analysis algorithm. Characteristics:

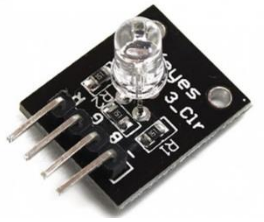

Figure 3.5: RGB LED Module [43]

- It has 4 pins: R for Red, G for Green, B for Blue, and a ground;
- It has 2V forward voltage for Red and 3.2V forward voltage for Green and Blue;
- As wavelengths, Red: 625nm, Green: 476.5nm, Blue:520nm;

3.3 CLOUD

3.3.1 NODE.JS ON GOOGLE CLOUD PLATFORM

Through the Google Cloud Platform (GCP) [39], Node.js [40] is used as a backend server environment for communication between the app, the database, and the device, used for handling sentiment analysis and computing the light colours. Because of its event-driven architecture, Node.js is perfect for real-time applications. It has an event-driven architecture, efficient for managing concurrent requests. By using GCP, the project benefits from scalable and secure backend infrastructure.

3.3.2 AFINN

Presented in Figure 3.6, using the AFINN-en-165 list of terms [6], which includes more than 3000 English words with corresponding provided score, the emotion of the user is evaluated, assigning a total score to complete phrases. AFINN is known for its effectiveness in processing text data, allowing forward calculation of the overall text's sentiment. It is fast and provides good accuracy, making it fit for this real-time application where speed is crucial.

```
hapless -2
haplessness -2
happiest    3
happiness   3
happy   3
harass  -3
harassed    -3
```

Figure 3.6: Words and Scores from AFINN-en-165 list

3.4 FIREBASE

3.4.1 REALTIME DATABASE

Firebase's [41] Realtime Database is used to store and sync data between the mobile app, the server, and the device. Being a NoSQL database, it allows efficient storage and retrieval of data, and its synchronisation capabilities offers immediate feedback, suited for the rapid exchange of information between the software and the hardware part.

3.4.2 AUTHENTICATION

To have secure user authentication, the mobile app uses Firebase Authentication, allowing users to create an account and log in to access their room analysis and their chat messages. This authentication system includes encryption to protect user data, maintaining privacy and security to the users. It accepts different methods of authentication, but this app uses the classic email/password sign up.

3.4.3 FIRESTORE

For long-term data storage, such as chat history or the sentiment analysis results of one user, the application uses Firebase's Firestore. Offering advanced querying capabilities, it is a flexible and scalable database, used for storing more complex data structures and queries.

3.5 GIT AND GITHUB

To ensure a good organisation and management of the development process, the source files of this project are stored using the provided tools by Git on the GitHub [42] platform. GitHub allows easy tracking of progress and changes overtime, giving full control over the history of every version of the project.

4. DESIGN

4.1 SYSTEM ARCHITECTURE

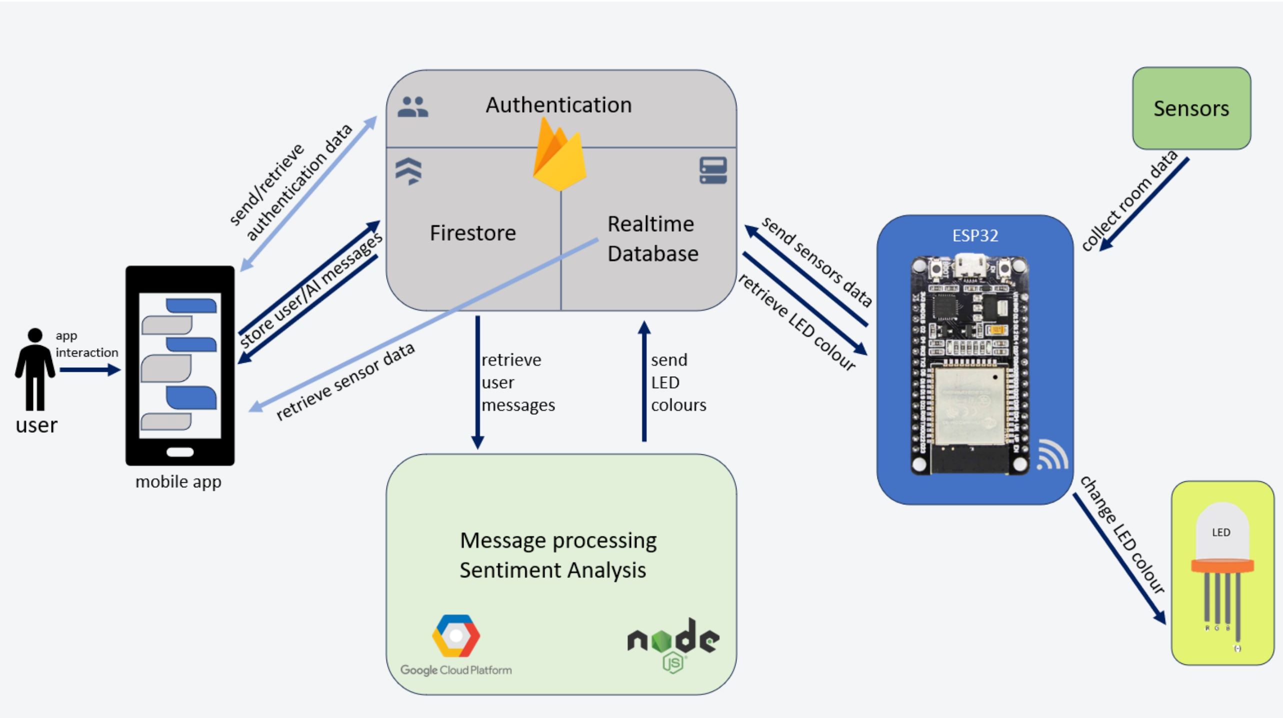

Figure 4.1: The System Architecture of MoodUp's System

To get the full experience of MoodUp, the user needs to install the mobile application as a first step. Here, the user is asked to log in or create a new account via Firestore's Authentication. After logging in, the user can access the functionalities of the app.

The second step for the user is to plug in the hardware device. This connects to the user's home wi-fi, reading sensor data the moment it senses motion. The room data is stored in the Realtime Database.

After the user sends a new message through the chatting part of the mobile app, an AI generated response will appear, both messages being saved in Firestore. The server on the Google Cloud Platform intercepts each message, performing the sentiment analysis algorithm that generates a score. This result is mapped to a colour and updates the Realtime Database. From here, the device reads the computed colour and changes the LED accordingly.

As the third step, to access the room data, the user must connect the device through the mobile interface using a unique device code. Once connected, they can monitor the environment and manually adjust the LED as they please.

The final step consists of accessing the mood analysis over the past few days, the user being able to see their stats via the profile page of the app. Logging out is straightforward once the user is done with the application.

Therefore, following these four big steps, the user can enjoy the full functionality of the Smart Ambient Lighting System.

The previously described workflow is classified as a Client-Server Architecture with Cloud Integration, the following main subsystem form MoodUp's System:

1. Mobile Application Subsystem
2. Hardware Device Subsystem
3. Cloud Analysis Subsystem

4.2 MOBILE APPLICATION SUBSYSTEM

MoodUp's mobile app serves as the primary interface for the user. Analysing the mobile application subsystem, each interaction that the user can have through the interface can be observed in Figure 4.2. As well, this diagram provides all the direct connections with the three database services provided by Firebase.

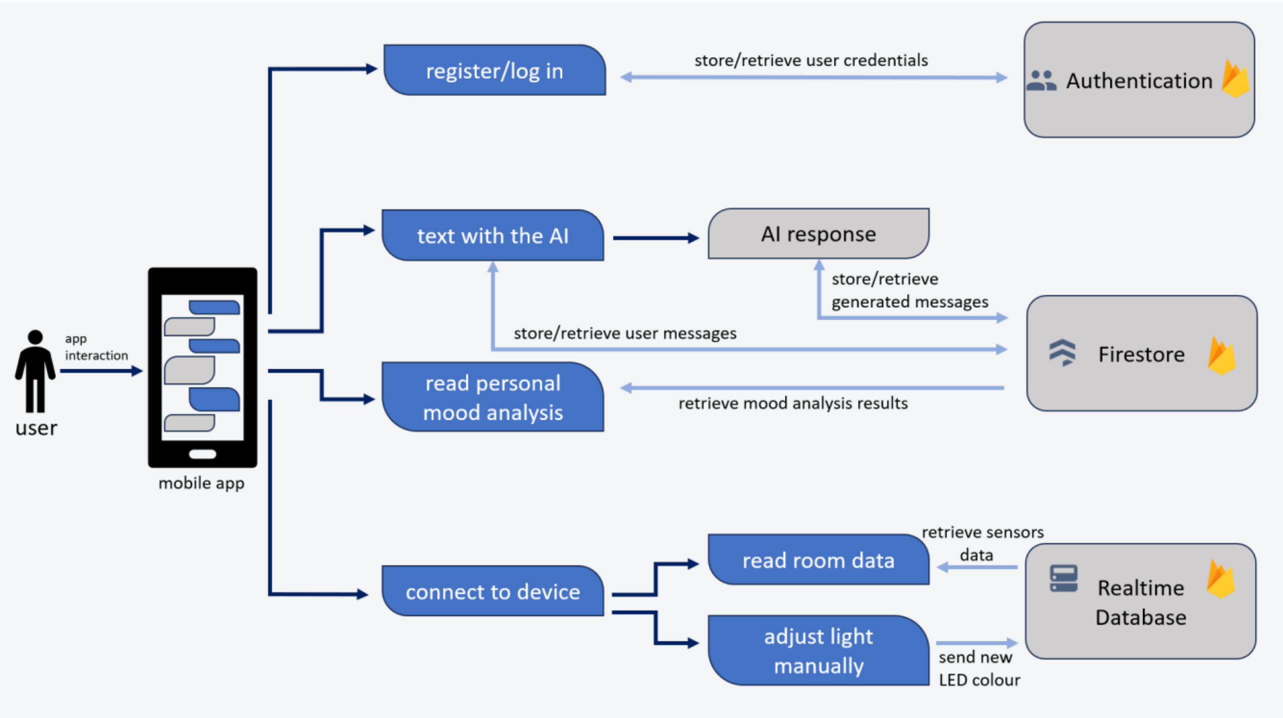

Figure 4.2: The Mobile App Subsystem of MoodUp's System

This system's application revolves around one single type of actor, which is the user. As presented in Figure 4.3, there are three main actions a user can make inside the application.

After creating an account and logging in, the user can start interacting with the application. They can text with the AI, can read their personal mood analysis, and can connect to their hardware device to monitor their room environment with the help of the data acquired from the Realtime Database.

The part of changing lights is performed once the user chats with the AI and the algorithm from the server analyses the sentiment of their text messages. The user also has the possibility to change the lights manually. Any of these changes will update the Realtime Database part of Firestore.

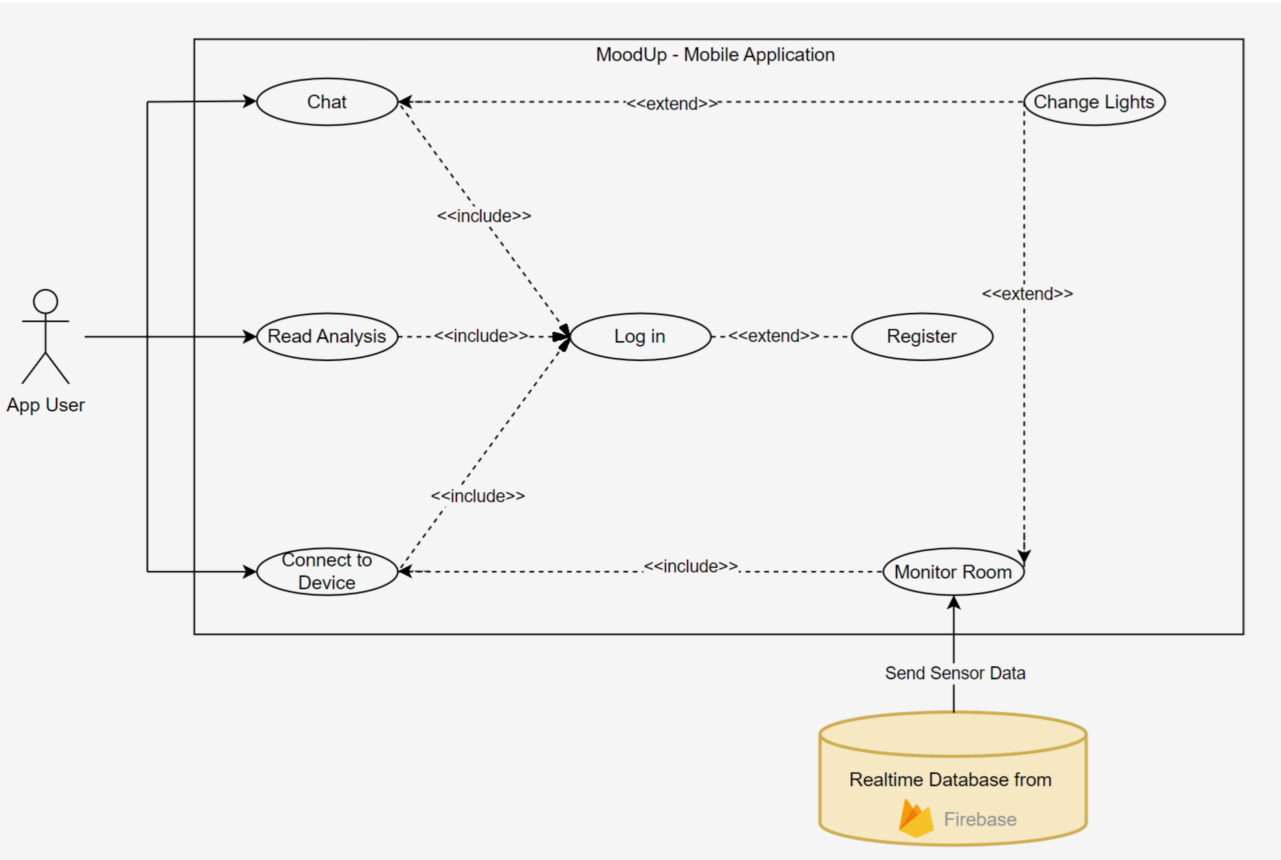

Figure 4.3: Use Case Diagram of MoodUp's Mobile App

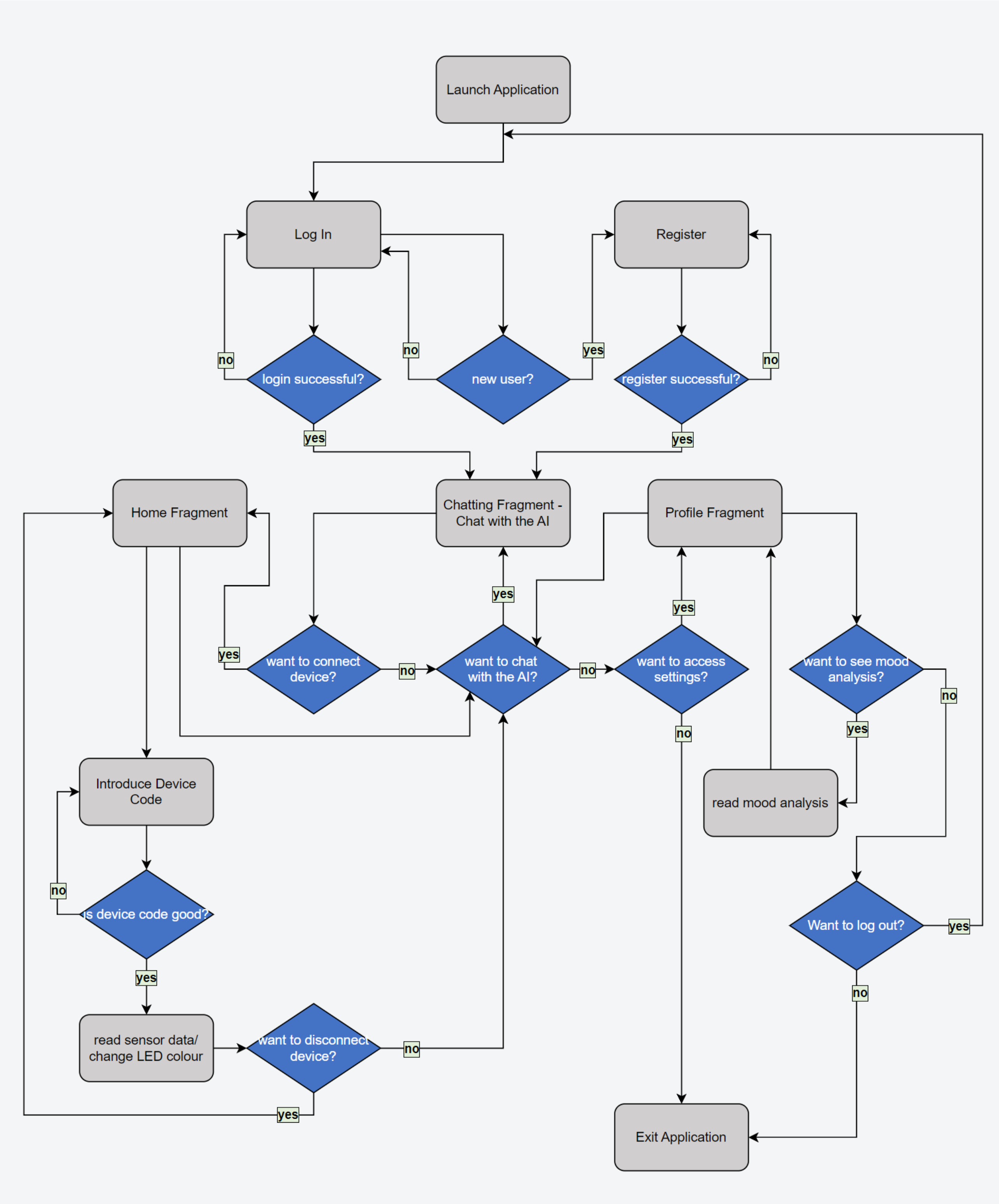

Figure 4.4: Workflow Diagram of MoodUp's Mobile App

To see the exact steps of how a user can interact with the app, a workflow diagram is used, presented in Figure 4.4

The process begins with the application launch. Here, the user is met with the Login Activity. If they are not a new user, they can proceed with logging in, redirected to the Chatting Fragment only if the signing in is successful.

The user can access the Register Activity if they are a new user. After creating an account and only if the registration is successful, they will be able to access the Chatting Fragment.

The user can start texting with the AI once they can access the Chatting Fragment, being able to see their chat history displayed on the screen. The user can switch immediately to the two other fragments, Home Fragment and Profile Fragment, depending on their liking.

If the user wants to connect to the hardware device, they can access the Home Fragment. Here, they introduce the device code and can continue only if the provided code is valid. Once connected, the user can read the room analysis data, or they can manually change the colour of the LED.

The user is presented with the option of disconnecting the device, which would bring them back to the initial view of the Home Fragment. But if the user wants to keep the connection as it is, they have the option to go back to one of the other two fragments, Chatting Fragment or Profile Fragment.

If the user does not want to continue chatting, they can switch to the Profile Fragment, where they have access to their mood analysis. Once they do that, they can return to the initial view of the Profile Fragment. The user also has the option to log out of the application, which would bring them back to the Login Activity.

And after all these steps, if the user does not want to perform any other action within the mobile app, they have the option to exit the application, bringing the workflow to an end.

As such, the workflow of the mobile application can be followed from beginning to the end in a simple and straightforward way.

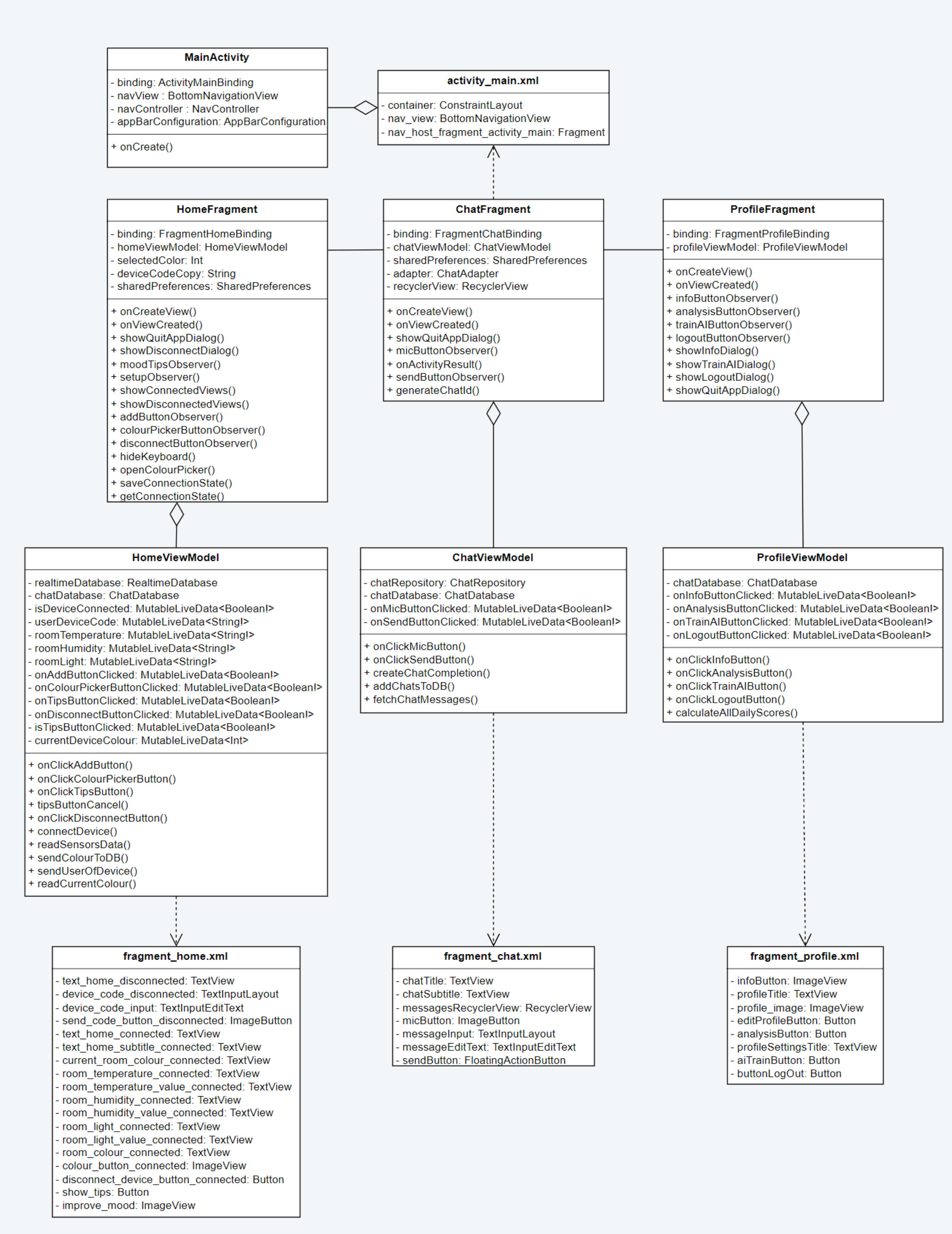

Figure 4.5: UML Class Diagram of MoodUp's Mobile App for the Main Activity and Its Bottom Navigation Fragments

As it is illustrated in Figure 4.5 with the help of the UML Class Diagram of MoodUp's Mobile App, it can be observed the structure and relationship between the MainActivity and its associated Bottom Navigation Fragments.

The diagram is organised according to the Model-View-ViewModel architecture. The entry point of the app is given by the MainActivity that manages the BottomNavigationView, presenting an uncomplicated way of navigating between the HomeFragment, ChatFragment, and ProfileFragment. Each fragment has its own ViewModel that prepares data for the user interface and manages the business logic of each fragment.

4.3 HARDWARE DEVICE SUBSYSTEM

The hardware device consists of different parts that form the hardware device subsystem of MoodUp, as presented in Figure 4.6.

The device has a main microcontroller integrated circuit present in the ESP32 board, its integrated wi-fi connection enabling direct connectivity to Firebase's services. There are three sensor modules connected to it, a Temperature and Humidity Sensor Module, a Light Sensor Module and a Motion Sensor Module, and an RGB LED Module.

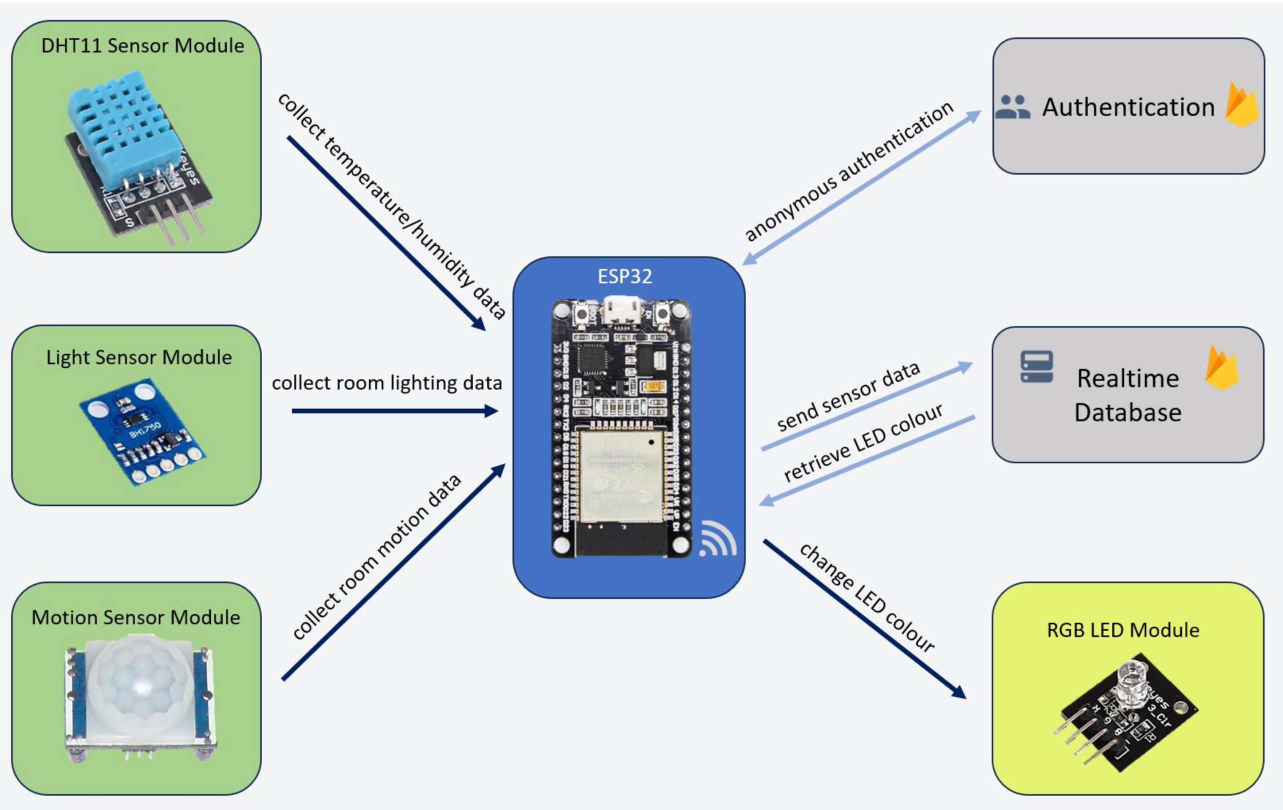

Figure 4.6: The Hardware Device Subsystem of MoodUp's System

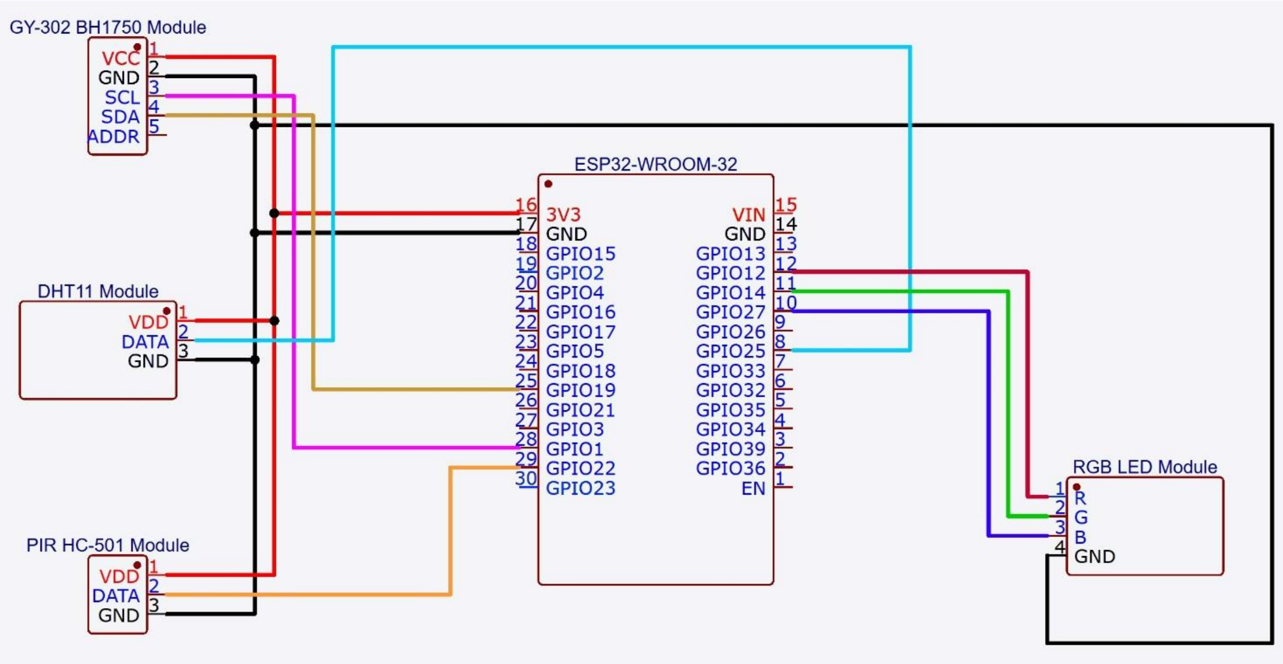

Figure 4.7: The Hardware Architecture of MoodUp's Device

Illustrating the interconnections between the four components and the ESP32 microcontroller, Figure 4.7 represents the hardware architecture of the MoodUp device.

Using a I2C protocol that requests two wires, SCL and SDA, transmitting data efficiently with minimal pin usage, the **GY-302 BH1750 light sensor module** has 5 pins, connected as follows:

1. VCC pin to the 3.3V pin of the ESP32, powering the sensor;
2. GND pin to the ground of the ESP32;
3. SCL (Serial Clock Line) pin to the GPIO22 of the ESP32;
4. SDA (Serial Data Line) pin to the GPIO21 of the ESP32;
5. ADDR pin not connected, it is optional for this setup.

Using a single-wire digital protocol for minimal wiring, the **DHT11 temperature/humidity sensor module** has 3 pins, connected as follows:

1. VDD pin to the 3.3V pin of the ESP32, powering the sensor;
2. DATA pin connected to the GPIO14 of the ESP32;
3. GND pin connected to the ground of the ESP32.

Having a directly read digital input, the **PIR HC-SR501 motion sensor module** has 3 pins, connected as follows:

1. VDD pin connected to the 3.3V pin of the ESP32, powering the sensor;
2. DATA pin connected to the GPIO14 of the ESP32;
3. GND pin connected to the ground of the ESP32.

Controlled using the pulse width modulation signal from the board, the *RGB LED module* has 4 pins, connected as follows:

1. R pin connected to the GPIO13 of the ESP32;
2. G pin connected to the GPIO12 of the ESP32;
3. B pin connected to the GPIO27 of the ESP32;
4. GND pin connected to the ground of the ESP32.

The integration of these components can be observed in Figure 4.8, illustrating a possible representation of the physical device’s wiring. As the ESP32 microcontroller isn’t breadboard friendly, it needs a custom-board. One can be created by putting together a power rail and two tiny boards. Lastly, various colourful wires are used for better differentiation between pins. Red represents the powering that is connected to the 3.3V of the ESP32, and black represents the ground of the microcontroller.

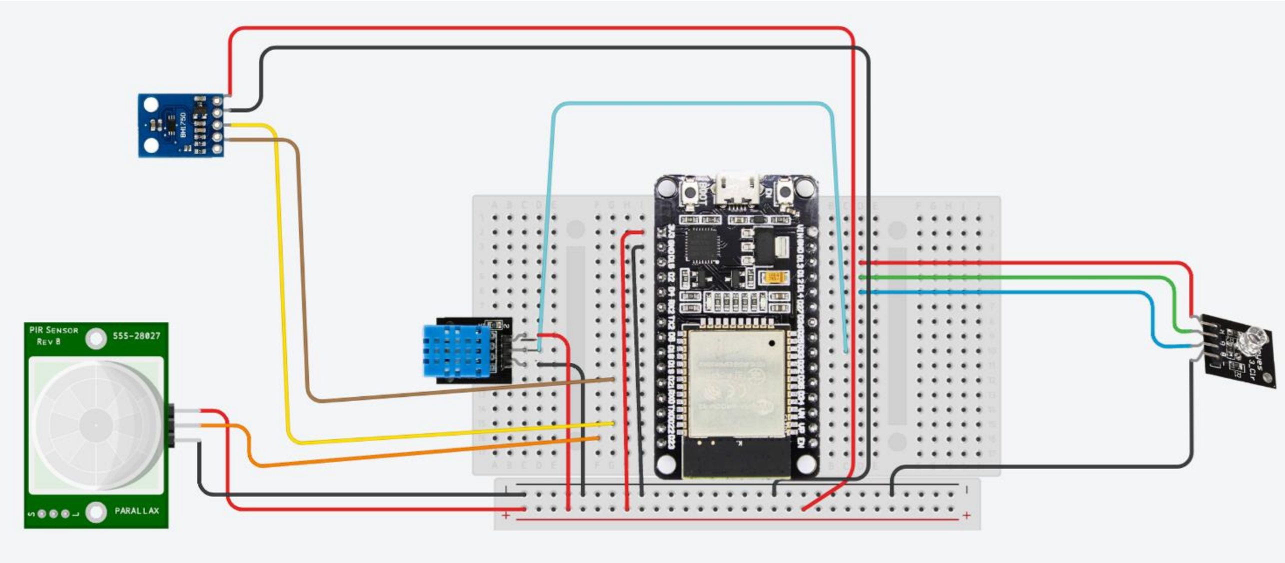

Figure 4.8: Hardware Wiring of MoodUp's Physical Device

4.4 CLOUD ANALYSIS SUBSYSTEM

Using a Node.js backend through the Google Cloud Platform, Figure 4.9 depicts an easy representation of the algorithm logic that takes care of the messages' sentiment analysis.

The algorithm finds the current connected user from the Realtime Database and retrieves from Firestore the messaging history of the user. Then, using a custom-made sentiment analysis function, these messages get assigned a score based on the overall sentiment of the text. This score is then mapped to a specific colour which is sent to the Realtime Database, so the device can change the colour of the LED. Lastly, the score of the messages is pushed to Firestore for further analysis in time of the overall mood of the user.

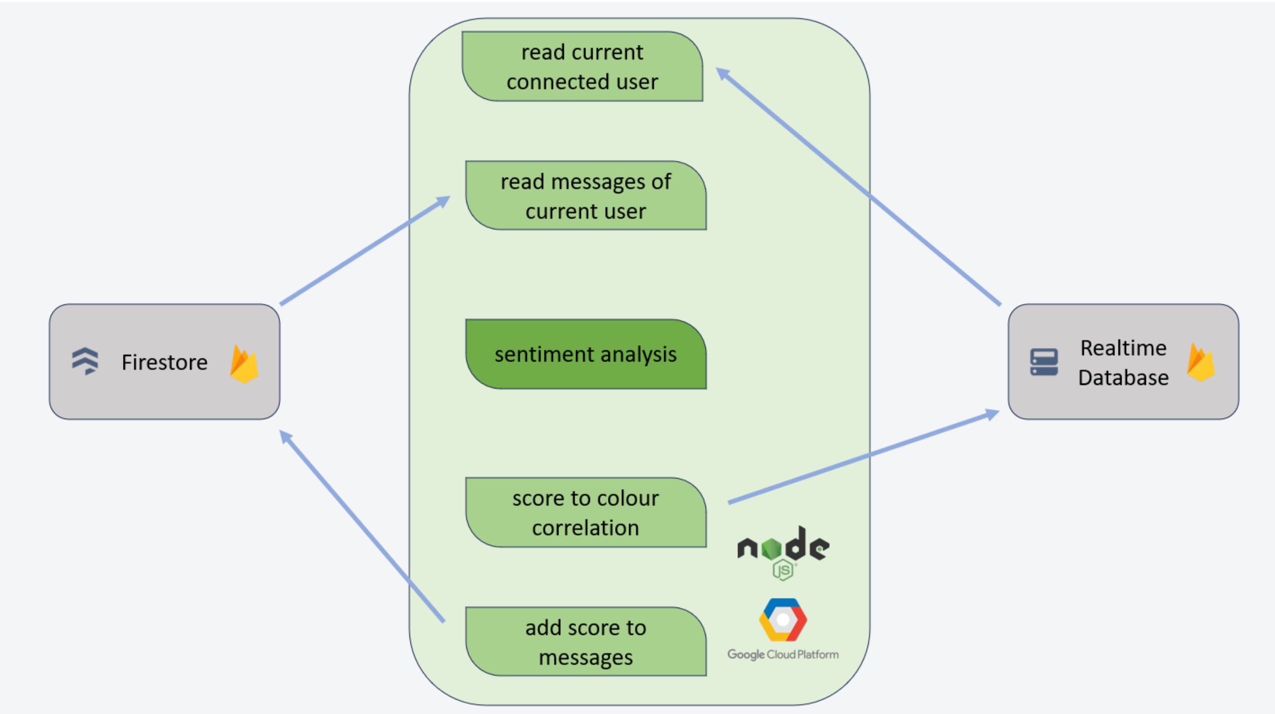

Figure 4.9: The Cloud Analysis Subsystem of MoodUp's System

5. IMPLEMENTATION

5.1 OVERALL IMPLEMENTATION

The implementation of this thesis project can be broken down into three main components, as follows:

1. Software: the implementation of MoodUp's mobile application;
2. Hardware: the implementation of MoodUp's hardware device;
3. Cloud: the implementation of MoodUp's sentiment analysis.

5.2 SOFTWARE – MOBILE APPLICATION

The mobile application is developed in the Android Studio IDE using Kotlin and follows the Model-View-ViewModel (MVVM) architecture. The implementation is broken down into different Activities and Fragments which together compose the entire project.

5.2.1 LOGIN AND REGISTER ACTIVITIES

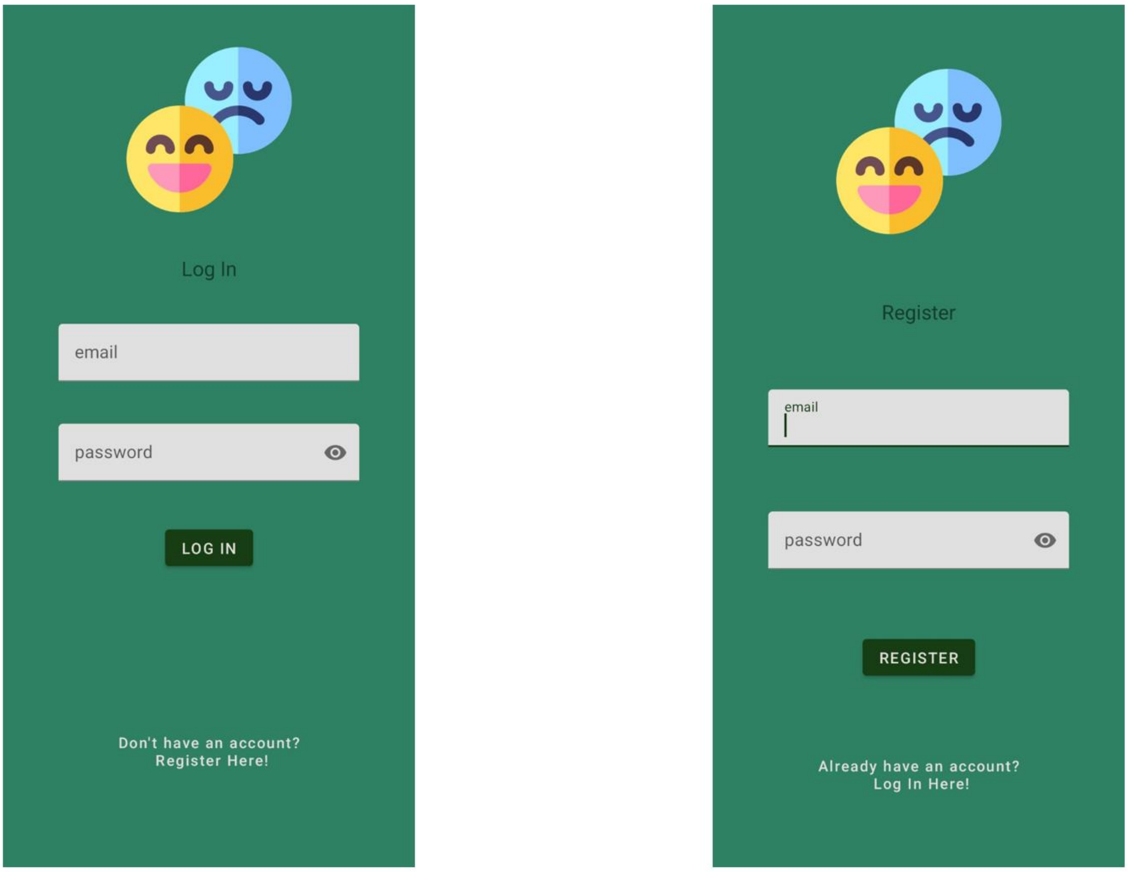

Figure 5.1: activity_login and activity_register of MoodUp's Mobile Application

The Login and Register functionalities share near the same implementations as they have similar purposes. They are both implemented using the provided authentication methods present in the FirebaseAuth class, namely ‘signInWithEmailAndPassword’ and ‘createUserWithEmailAndPassword’, so to avoid writing the same thing twice, the demonstration of the coding workflow will be done only for the Registration part.

Depicted in Figure 5.2, the user can introduce their email and create a password, then press the ‘REGISTER’ button. These fields have the standard Firebase Authentication restrictions, like having a valid email address or needing a password longer than 6 characters.

Inside the ViewModel part, respectively the RegisterViewModel class, the Code Snipped 1 shows how two MutableLiveData String values are declared for the user input email and password. This allows the data to be updated automatically in real time if the user decides to make any changes, lowering the possibility of getting errors. Then, the ‘onRegisterButtonClicked’ control value is declared, its purpose being to make sure that the registration request is sent only when the user presses the ‘REGISTER’ button, its value updated by the function ‘onClickRegisterButton’.

```
val registerEmail = MutableLiveData( value: "")
val registerPassword = MutableLiveData( value: "")
val onRegisterButtonClicked = MutableLiveData( value: false)
fun onClickRegisterButton() {
    onRegisterButtonClicked.value = true
}
```

Code Snippet 1: User Input Data Preparation from RegisterViewModel Class

As shown in Code Snippet 3, the final step is to create an observer function for the ‘REGISTER’ button in the RegisterActivity class, based on the previously mentioned control variable, ‘onRegisterButtonClicked’.

When its value becomes true, the user input data will be checked for errors, and, repeating the process if there are any. When the data is correct, the authentication process starts by calling the ‘createUserWithEmailAndPassword’ method on ‘moodUpAuth’ variable. If the authentication succeeds a Toast message will be shown to confirm and the view will change from activity_register to activity_main. If the registration fails, an error will be shown on the user’s screen.

```kotlin
private fun registerButtonObserver() {
    registerViewModel.onRegisterButtonClicked.observe( owner: this) { isClicked ->
        if (isClicked) {
            if (registerViewModel.registerEmail.value.isNullOrBlank()) {
                Toast.makeText( context: this@RegisterActivity,  text: "Enter email", Toast.LENGTH_SHORT).show()
                return@observe }
            if (registerViewModel.registerPassword.value.isNullOrBlank()) {
                Toast.makeText( context: this@RegisterActivity,  text: "Enter password", Toast.LENGTH_SHORT)
                    .show()
                return@observe }
            val rEmail: String = registerViewModel.registerEmail.value!!
            val rPassword: String = registerViewModel.registerPassword.value!!
            moodUpAuth.createUserWithEmailAndPassword(rEmail, rPassword)
                .addOnCompleteListener(this) { task ->
                    if (task.isSuccessful) {
                        val intent = Intent( packageContext: this@RegisterActivity, MainActivity::class.java)
                        startActivity(intent)
                        finish()
                    } else {
                        Toast.makeText( context: this@RegisterActivity,  text: "Authentication failed.", Toast.LENGTH_SHORT).show()
                    }
                }
            registerViewModel.onRegisterButtonClicked.value = false
        } } }
```

Code Snippet 2: registerButtonObserver() Function from RegisterActivity Class

5.2.2 MAIN ACTIVITY

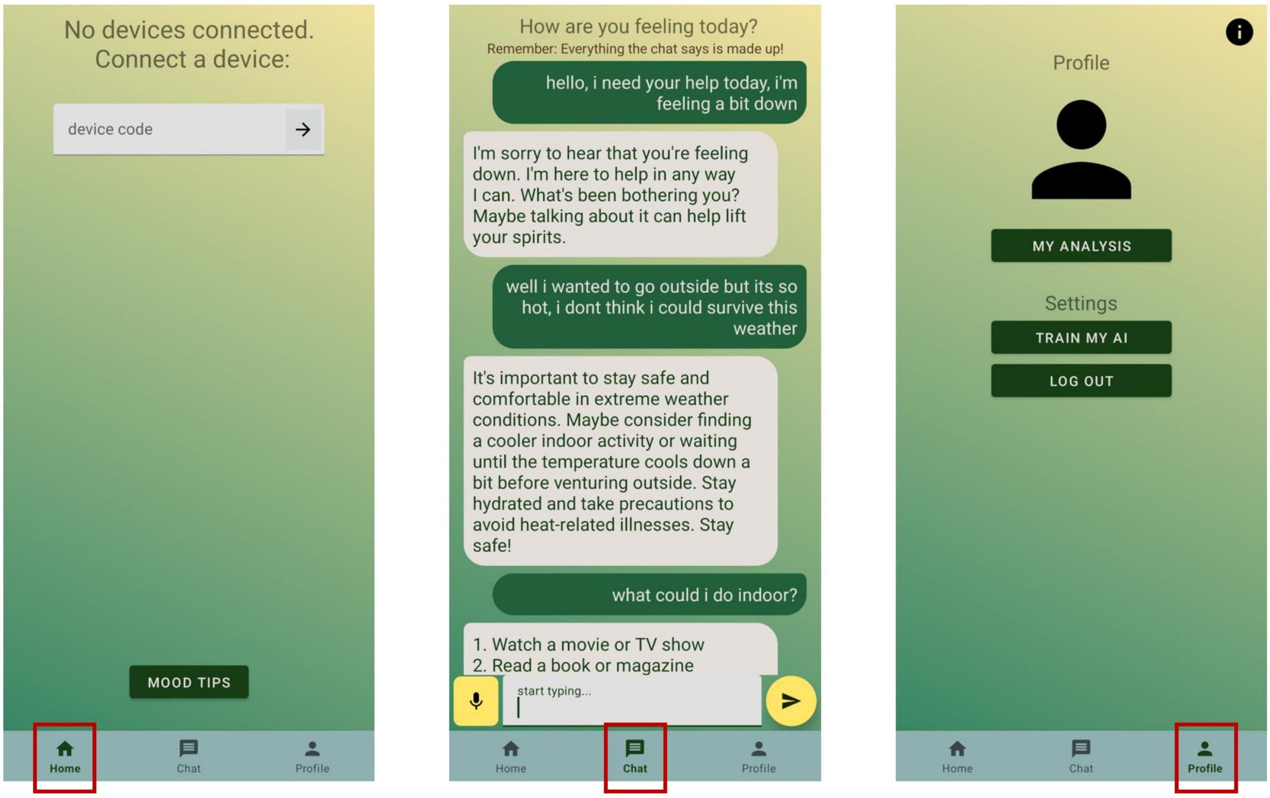

Figure 5.2: The Three Fragments that Form activity_main

The MainActivity takes care of setting up the bottom navigation, which serves as an intuitive method of navigation between the three main fragments of the application, namely fragment_home, fragment_chat, and fragment_profile, their interface shown in Figure 5.3.

5.2.2.1 HOME FRAGMENT

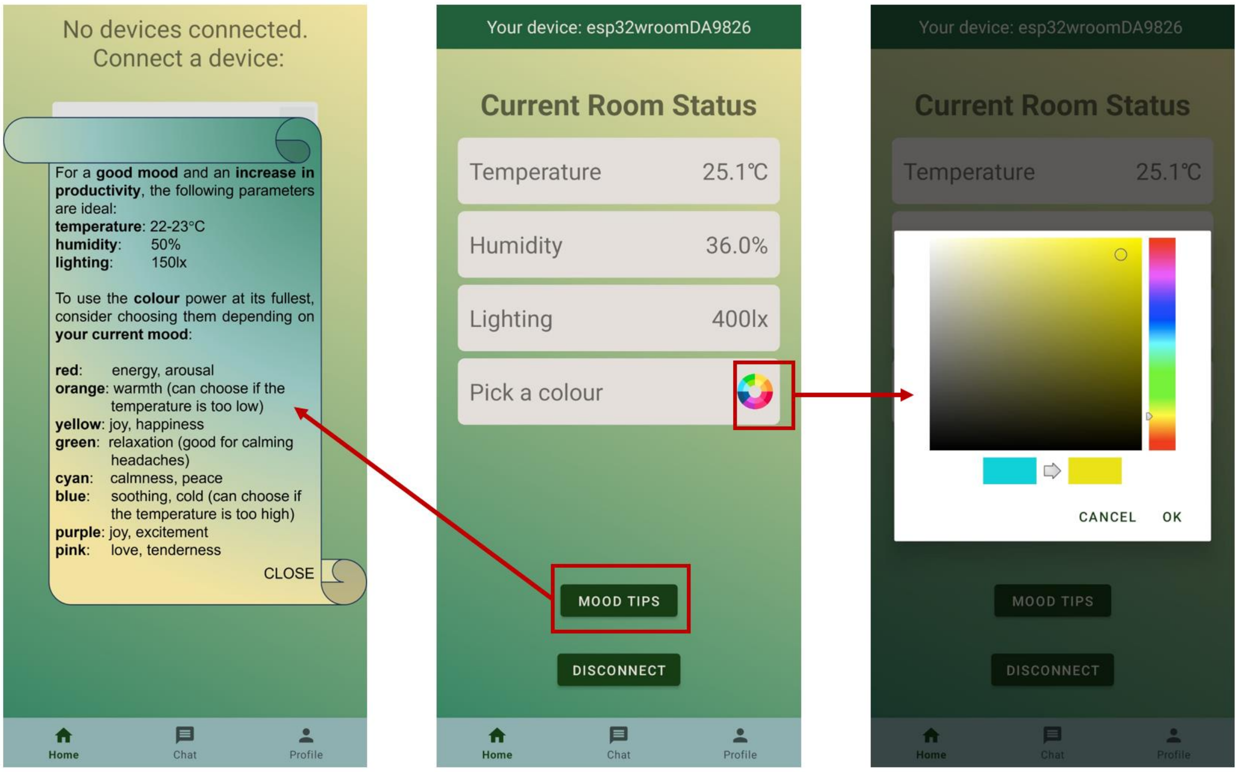

Figure 5.3: Actions that Can Be Done from fragment_home

Inside the fragment_home, the user can connect to a device to see the room status or change the LED light and find a few tips useful for a better state of mind, as presented in Figure 5.2.

The current room status is displayed after fetching the sensor data from the Realtime Database, all its logic stored inside the RealtimeDatabase class. The fragment_home has access to this data through the HomeViewModel class.

As presented in Code Snippet 3, the function ‘readSensorsData’ is created, which begins by referencing the path to the device’s sensor data within the Realtime Database using the ‘deviceCode’. To monitor the changes in the ‘sensors’ node, it makes use of the ‘addEventListener’, so the changes are updated automatically after every change. Inside the ‘onDataChange’ method, the function reads each sensor value from the snapshot, then the retrieved values being passed to the ‘onDataRecieved’ callback from where HomeViewModel can access the collected data. There is also an ‘onCancelled’ method to ensure the handling of any unexpected errors.

```
fun readSensorsData(deviceCode: String, onDataReceived: (String, String, String, Int) -> Unit, onError: (String) -> Unit
) {
    val sensorsRef = realtimeRef.child(deviceCode).child( pathString: "sensors")
    sensorsRef.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            val temperature = snapshot.child( path: "temperature").getValue(Float::class.java) ?: 0.0f
            val humidity = snapshot.child( path: "humidity").getValue(Float::class.java) ?: 0.0f
            val light = snapshot.child( path: "light").getValue(Int::class.java) ?: 0
            val motion = snapshot.child( path: "motion").getValue(Int::class.java) ?: 0
            onDataReceived(temperature.toString(), humidity.toString(), light.toString(), motion)
        }
        override fun onCancelled(error: DatabaseError) {
            onError("Failed to read sensor data: ${error.message}")
        } }) }
```

Code Snippet 3: readSensorsData() Function from RealtimeDatabase Class

5.2.2.2 CHAT FRAGMENT

The refresh of the messages displayed on the interface of fragment_chat is dependent on the ‘Send’ button, so an observer function is declared inside the ChatFragment class, called ‘sendButtonObserver’, as shown in Code Snippet 4. Here, an observer is attached to its control variable, ‘onSendButtonClicked’, declared inside the ChatViewModel class as a MutableLiveData Boolean.

After the user writes a message and the ‘Send’ button is pressed, the validation of the text is checked, then it gets assigned a locally generated chat ID to be added to Firestore using the ‘addChatsToDB’ function created inside the class ‘ChatDatabase’.

Then, based on the contents of the text, the ‘createChatCompletion’ function that is called through ChatViewModel from the ChatRepository class generates the AI responses with the help of a gpt-3.5-turbo API. This message is then added to the database inside the same class.

```
private fun sendButtonObserver(adapter: ChatAdapter, recyclerView: RecyclerView) {
    chatViewModel.onSendButtonClicked.observe(viewLifecycleOwner) { isClicked ->
        if (isClicked) {
            val messageInput: String = chatViewModel.messageInput.value!!.trim()
            if (messageInput.isNotEmpty()) {
                val chatId = generateChatId()
                chatViewModel.addChatsToDB(chatId, messageInput,  type: "sender")
                chatViewModel.messageInput.value = ""
            } else { Toast.makeText(context,  text: "please write something first!", Toast.LENGTH_LONG).show() }
            val chatId = generateChatId()
            chatViewModel.createChatCompletion(messageInput, chatId) } }
    chatViewModel.chatList.observe(viewLifecycleOwner) { it: List<Chat>!
        adapter.submitList(it)
        recyclerView.smoothScrollToPosition(it.size) }
    chatViewModel.onSendButtonClicked.value = false }
```

Code Snippet 4: sendButtonObserver() function from ChatFragment Class

Lastly, an observer is assigned to chatList that takes care of refreshing the view whenever a new message is added with the help of an adapter and a recyclerView, the results of this entire process visible in Figure 5.4.

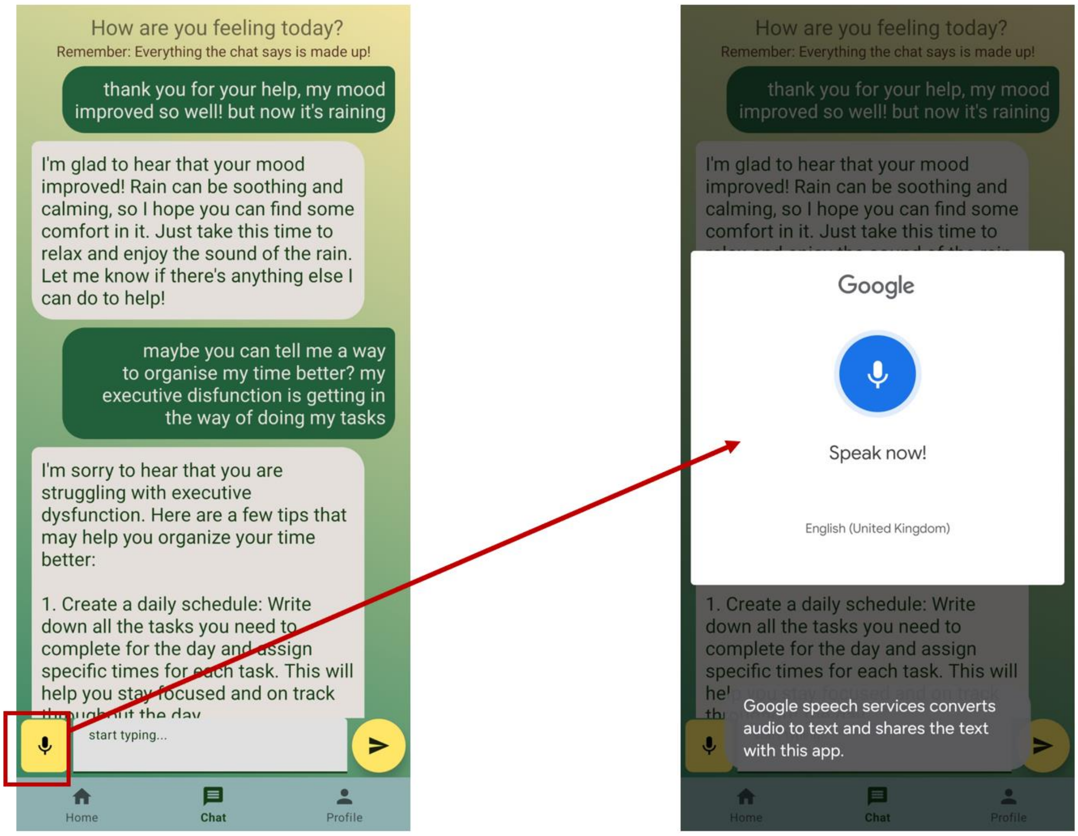

Figure 5.4: Actions that Can Be Done from fragment_chat

5.2.2.3 PROFILE FRAGMENT

Inside the ChatDatabase Class, which is responsible for the entire logic of the message storage and direct connection to Firestore, the function 'calculateAllDailyScores' is created. Here, after fetching all the messages of a user depending on the day, an average score will be assigned for the sentiment analysis results, filtered after positive and negative values, as presented in Code Snippet 5.

```
val finalDailyScoresMap = mutableMapOf<String, Map<String, Double>>()
for ((date, scoresList) in dailyScoresMap) {
    val positiveScores = scoresList.filter { it > 0 }
    val negativeScores = scoresList.filter { it < 0 }
    val positiveSum = positiveScores.sum()
    val negativeSum = negativeScores.sumOf { it.absoluteValue }
    val totalScores = positiveSum + negativeSum
    val positivePercentage = if (totalScores > 0) (positiveSum * 100 / totalScores) else 0.0
    val negativePercentage = if (totalScores > 0) (negativeSum * 100 / totalScores) else 0.0
    val dailyScore = mapOf("positivePercentage" to positivePercentage, "negativePercentage" to negativePercentage)
    finalDailyScoresMap[date] = dailyScore }
```

Code Snippet 5: Calculation of the Percentage Scores from ChatDatabase Class

To determine the percentage scores, all the results of the sentiment analysis fetched for one day are stored in 'dailyScoresMap'. This map is sought through, filtering the data based on positive and negative values. Then, the average for each is computed and transformed to fit the right percentage, these results being pushed back to Firestore.

This Function will be called whenever the user presses the 'MY ANALYSIS' button, so the update will be constant as new data keeps being pushed. Then, inside the AnalysisViewModel, these calculated percentages are retrieved and displayed in a bar chart, the results visible in Figure 5.5.

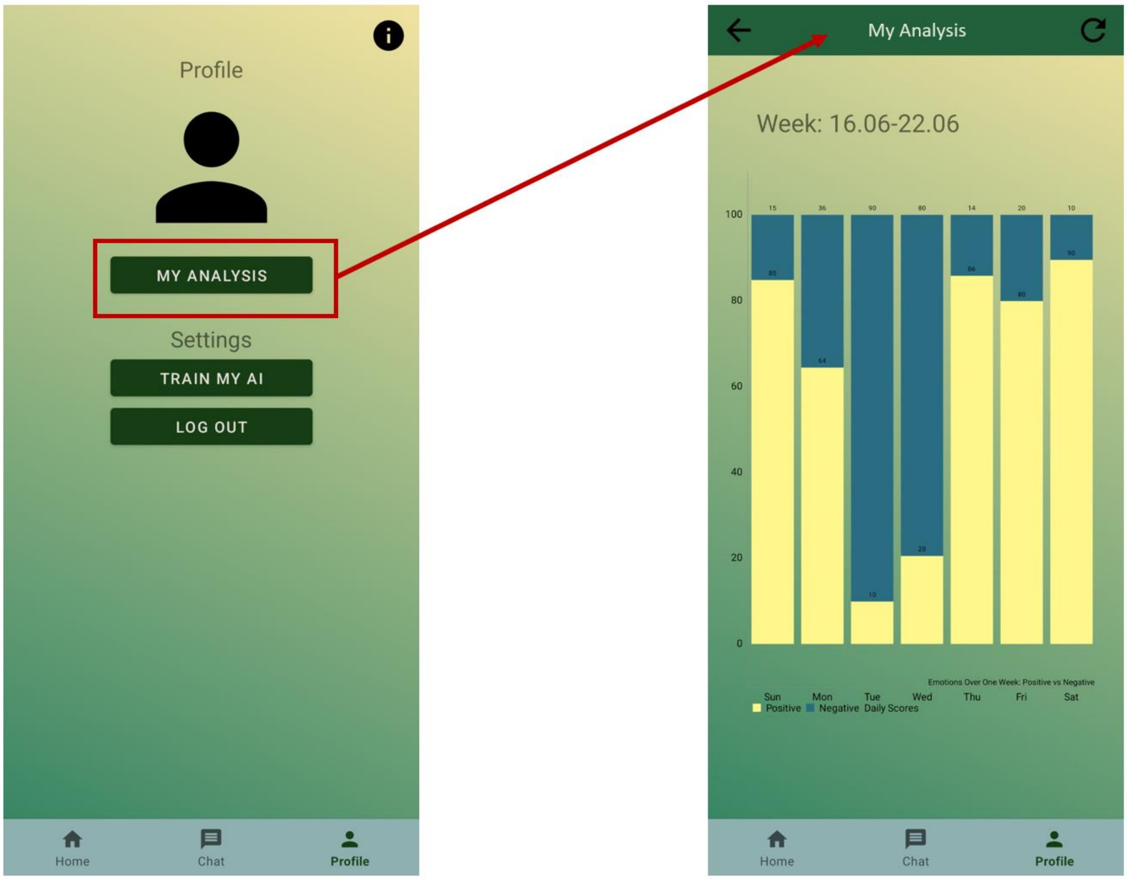

Figure 5.5: Actions that Can Be Done from fragment_profile

5.3 HARDWARE – DEVICE

With the help of the Arduino IDE and the C++ programming language, the code uploaded on the ESP32 microcontroller is intuitive and is easy to follow. Its workflow starts from connecting the board to Wi-Fi and Firebase, reading data from various sensors, then changing the LED light after fetching its colours from the Realtime Database.

To understand the logic of the loop() function, it is necessary to observe how some of the support functions are implemented.

After the wi-fi and Firebase connections are established, using the help of 'Wifi.h' and 'Firebase_ESP_Client.h' libraries, the parameters of the room are read using the sensors.

One example is the PIR sensor reading function, shown in Code Snippet 6. The PIR sensor state is directly read through a digital input. If it reads high, the motion variable is set to 1, but if it reads low, the motion is set to 0. This way, the motion sensor acts as a switch.

```
void readPIRSensor(){
  int motion = digitalRead(PIR_PIN);
  if (motion == HIGH)
  {
    intMotion = 1;
    Serial.println("Motion detected!");
  }
  else
  {
    intMotion = 0;
    Serial.println("No motion!");
  }
}
```

Code Snippet 6: Function that Reads PIR Sensor

All the read sensor data, namely the temperature, humidity, light, and motion of the sensor data, are pushed to the Realtime Database. The LED light results are also taken from here, each colour value is read, R for Red, G for Green, respectively B for blue, a representation of how colours are created in RGB base being shown in Figure 5.6.

It is necessary to consider that the LED has 3 pins, each corresponding to the previously mentioned primary colours, the final colours being the resulting combination of these three. This is achieved in code by mapping the integer values for each colour, ranging from o to 255, to pulse-width modulation (PWM) values, ranging from 0 to 1023.

```
void setColourToLED(){
  delay(100);
  int pwmRed = map(intRed, 0, 255, 0, 1023);
  int pwmGreen = map(intGreen, 0, 255, 0, 1023);
  int pwmBlue = map(intBlue, 0, 255, 0, 1023);

  analogWrite(RED_PIN, pwmRed);
  analogWrite(GREEN_PIN, pwmGreen);
  analogWrite(BLUE_PIN, pwmBlue);
}
```

Code Snippet 7: Function that Sets the Colour to the LED

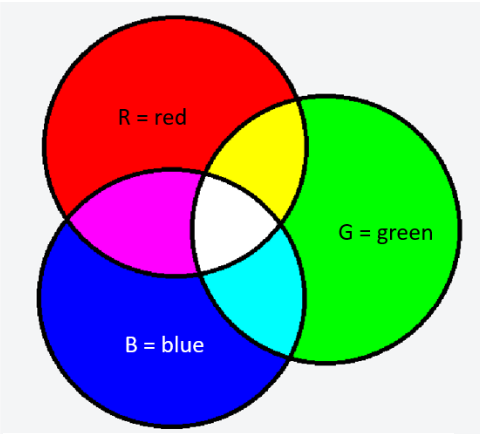

Figure 5.6: How RGB Base Colours Are Formed

As presented in Code Snippet 7, this is made possible by using the 'map' function that converts the colour intensity values to PWM ones. Then, using a direct write as analog output, these values are applied to the corresponding LED pins.

```
loop:
    if Firebase is ready and signup is OK:

        if motion is not detected:
            read PIR sensor and send motion data to Firebase
            if motionDetected is 1:
                record current time as lastMotionRead

        if motion is detected and either enough time has passed or it's the first data send:
            update sendDataPrevMillis with current time
            read light sensor and read DHT sensor and send sensor data to Firebase
            read color data from database and set LED color
            read PIR sensor and send motion data to Firebase
            if motion is detected:
                record current time as lastMotionRead

            if no motion has been detected for more than 20000 milliseconds:
                read PIR sensor and send motion data to Firebase
                if motion is still not detected:
                    set LED color to black (off)
                    delay for 5000 milliseconds (to prevent false motion readings)
                    set motionDetected to 0 (false)
```

Code Snippet 8: Pseudocode of the loop() Function

To address the main functionality of the loop() function, the pseudocode presented in Code Snippet 8 illustrates the walk-through of the algorithm. If the Firebase connection is set and ready, the motion detected by the PIR sensor is read, continuously reading until there is movement in the room.

When the motion is detected, the light and DHT sensors are read in a 2 second repeated interval, and their data is pushed to Firestore, the changes being updated in real time. Then the PIR sensor is read, updating the time of each motion detected.

If there is no movement in the room over a longer period, namely 20 seconds, the PIR sensor is read again to ensure no errors are made, then the light is turned off and the motion control variable is set to 0, so the loop can check it when it starts over again.

5.4 CLOUD - SENTIMENT ANALYSIS

The backend server that runs on Google Cloud Platform using Node.js takes care of the sentiment analysis of the user input messages. Its workflow starts with reading the current connected user then reading their messages, performing the sentiment analysis, mapping the results to a specific colour, then pushing it to the Realtime Database and the analysis results to Firestore.

To perform the sentiment analysis algorithm, there needs to be some preparations to be done. The implemented algorithm uses the AFINN-en-165 list, so the first step is to seek through the entire list to map each word to its individual provided score, done inside the ‘sentimentWords’ object. After all the messages are fetched, for each message the function ‘mySentimentAnalysis’ will be called, its implementation shown in Code Snippet 9.

```
function myTokeniser(message) {
  const cleanMessage = message.toLowerCase().replace(/[^a-z\s]/g, '');
  return cleanMessage.split(/\s+/);
}

function mySentimentAnalysis(message) {
  const tokens = myTokeniser(message);
  let score = 0;
  let counter = 0;
  tokens.forEach(token => {
    if (sentimentWords.hasOwnProperty(token)) {
      score += sentimentWords[token];
      counter++;
    } else {
      score += 0;
    }
  });
  if (counter > 0) {
    score = score / counter;
  } else {
    score = 0;
  }
  return score;
}
```

Code Snippet 9: Tokeniser and Sentiment Analysis Functions for Processing User's Messages

After the user's messages are cleaned using a custom tokeniser function that removes punctuation and replaces uppercase letters to lowercase ones, then splitting the message into individual words.

Then, 'mySentimentAnalysis' function continues by seeking through all the mapped words, increasing the value of 'score' and 'counter', both previously set to 0, whenever there is a match. After all the individual words of the message are checked, an average score is performed, returning it to the function that called the sentiment analysis one, so it can be sent to Firestore.

The score is represented by a float number that can get values from -1 to 1, so the colours are mapped based on different intervals, as presented in Figure 5.7, the score denoted with 'sc'. If the score is 0, meaning a neutral mod, the light is set to white. Then, the mapped colour is sent to the Realtime Database.

Figure 5.7: Colour Mapping for LED

6. TESTING

6.1 PRECONDITIONS

The testing part focuses on the method of Blackbox testing, that represents a method of testing without knowing the internal structure of the code, only the functionality of the application. To achieve that, the software and the hardware part need to be brought together and the backend cloud server to be started. There have been some tests created, each having a set test procedure with its equivalent expected results.

For the tests that will be addressed in the following subchapters, the default preconditions are represented by the following:

1. Software: the application needs to be running, an account to be created, having access to the activity_main view;
2. Hardware: the device needs to be powered and prepared to read motion;
3. Cloud: the backend server needs to be running.

6.2 TESTING MESSAGE GENERATION AND LIGHT COLOUR CHANGING

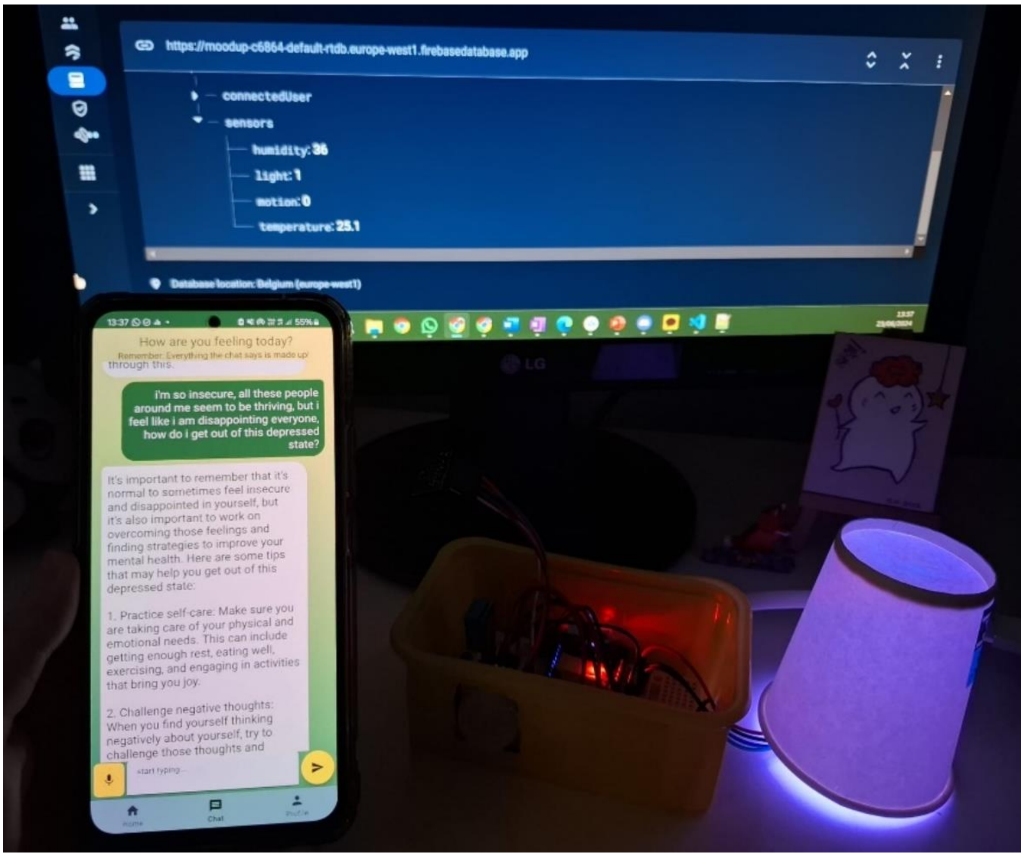

Figure 6.1: Testing the Scenario of Sending a Negative Message

Considering the default preconditions, the test can begin with navigating to the chatting fragment by pressing the Chat button from the bottom navigation. Now that the fragment has changed, a message with a negative feeling needs to be sent. The expected result of this is for the AI to generate a soothing message, and the colour of the device to be set to the right calming colour. In Figure 6.1 can be observed that these expected results are met.

Following the previous step, a message with a positive feeling needs to be sent, so the AI generates a message accordingly and changes the light to a joyful colour based on the level of excitement. The expected results are met and can be seen in Figure 6.2.

After proving that both previously mentioned conditions are met, the conclusion is that the test is passed.

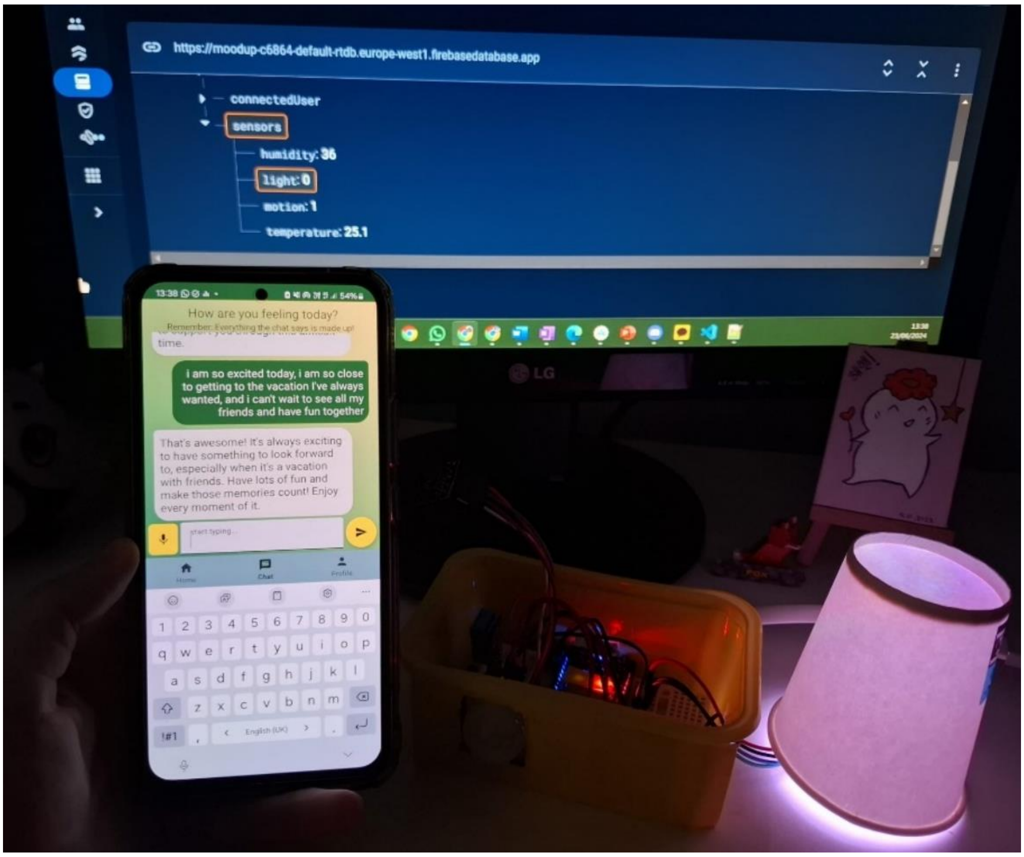

Figure 6.2: Testing the Scenario of Sending a Positive Message

6.3 TESTING DEVICE CONNECTION AND REALTIME UPDATE

Returning to the default preconditions, the first step is to navigate to the Home fragment and to introduce a valid device code. To test if the colour of the LED can be changed from the interface, the colour picker button is pressed, as presented in Figure 6.3. After choosing a colour, the 'OK' button is pressed. As well, the values of the current room status are expected to change as they appear in the Realtime Database.

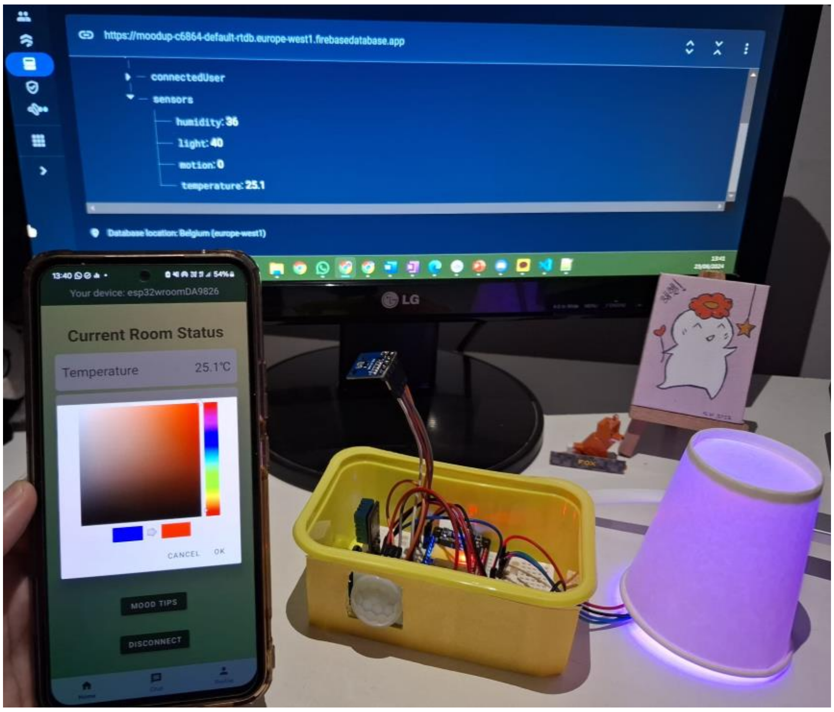

Figure 6.3: Testing the Scenario of Changing the Colour from the Mobile App

In Figure 6.4 can be observed that the LED has changed to the desired colour and that the interface has been updated with the Firebase values (visible on the computer screen). The conclusion is that this test is passed.

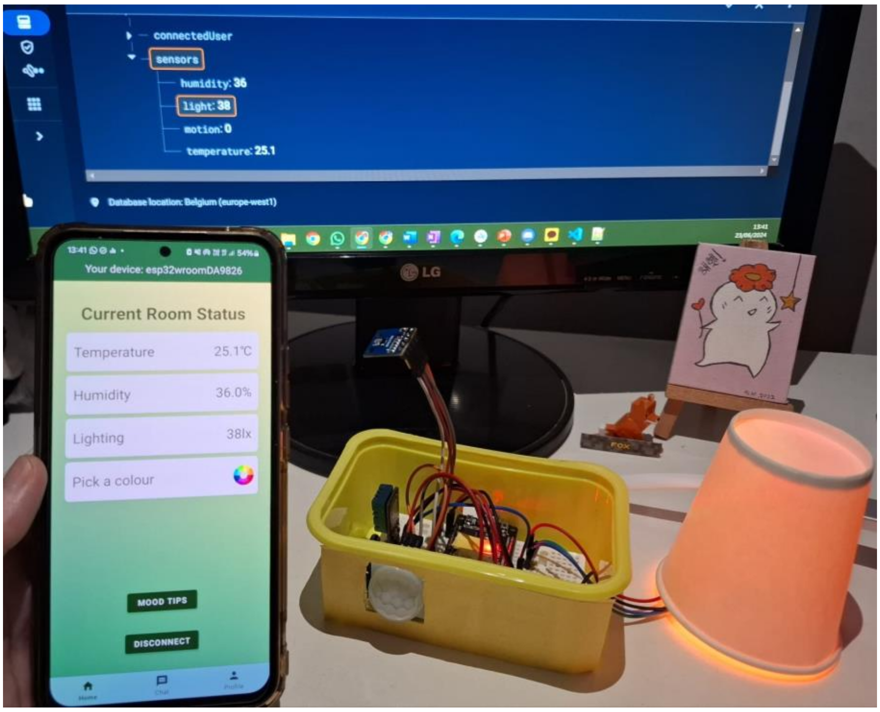

Figure 6.4: Testing the Scenario of Colour Change and Interface Data Update

6.4 TESTING SENTIMENT ANALYSIS UPDATE FOR NEW MESSAGE

As a last test, there needs to be checked if the sent new messages get an assigned sentiment score. To do that, following the default preconditions, the Chat Fragment is opened pressing the Chat button from the bottom navigation. Then, there will be sent only positive messages to see if the algorithm evaluates them accordingly, the expected values to be seen are only positive numbers.

This can be checked by opening the server console, as presented in Figure 6.5, as well by navigating in the mobile app to the Profile Fragment and pressing the 'MY ANALYSIS' button. Here a graph will be shown, presenting 100% positive messages. This proves that the proposed test is passed.

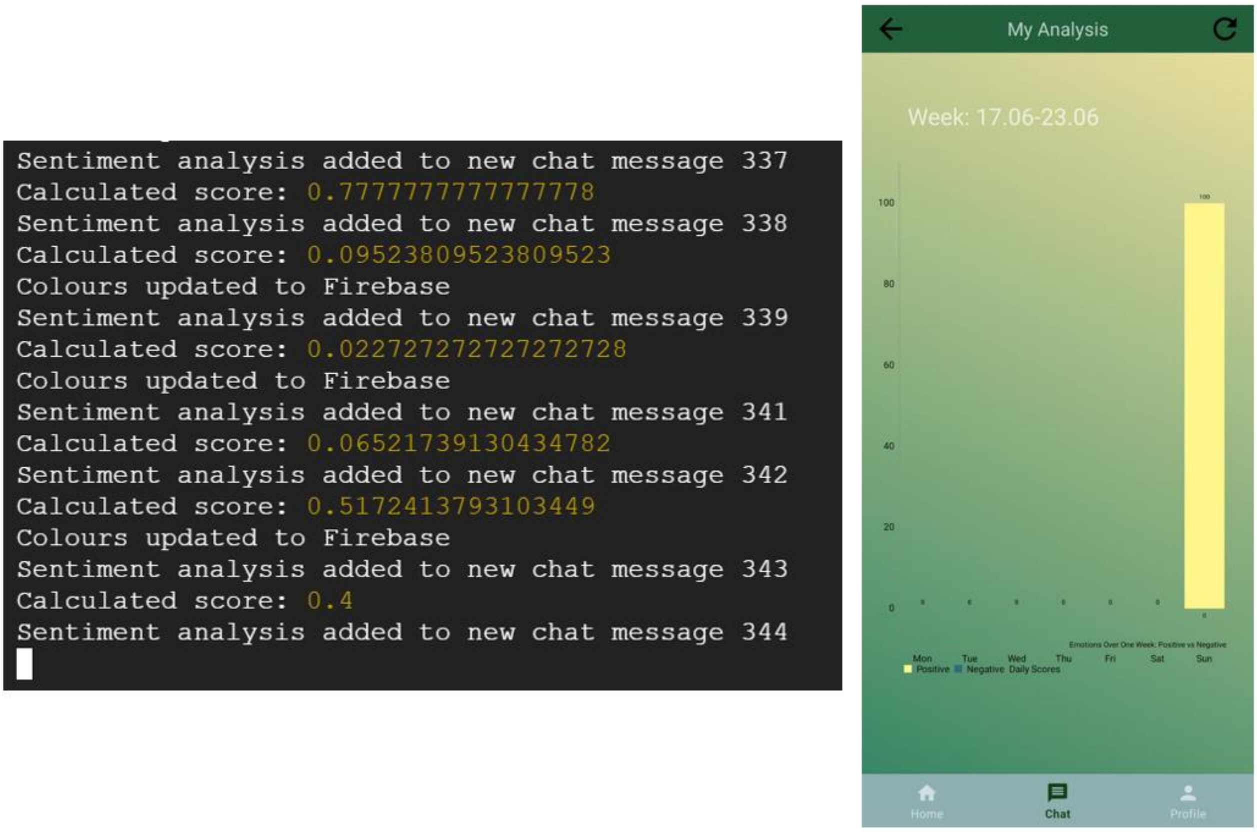

Figure 6.5: Testing the Scenario of Score Assignment to User's Message

7. CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSIONS

This thesis presents a Smart Ambient Lighting System designed to enhance the well-being of individuals by dynamically adjusting the lighting based on their mood. The system integrates a mobile app, hardware components and cloud services to provide a seamless user experience. Users can communicate with an AI assistant through the app, and their emotional state is analysed using the AFINN sentiment analysis algorithm. Based on the results, the lighting in their environment is adjusted to improve their mood and overall well-being.

Developed using Android Studio and Kotlin, the mobile app offers a user-friendly interface for interacting with the AI and monitoring environmental conditions. The hardware setup, featuring an ESP32 microcontroller, various sensors, and an RGB light module, collects environmental data and controls the lighting. Cloud services, including a Node.js server and Firebase, manage the data storage, synchronisation, and sentiment analysis.

The system demonstrates the potential of using technology to address mental health issues by creating a supportive and responsive environment.

7.2 FUTURE WORK

As any other project that exists, there have been identified a few areas of improvement and future development, considering both the hardware and the software part, which could enhance the system’s functionality and user experience. The end goal is that this system can act and react in a human-like manner.

7.2.1 SOFTWARE UPDATES

When it comes to software, there is always the desire to expand the product towards as many platforms as possible. Thus, the first step will be coming up with an IOS version of the application. Afterwards, there will be implementations compatible with smart cars, watches, etc. and smart homes computers.

The AI assistant will go through some changes, such as it will be able to provide more nuanced and context-aware responses. The user will have complete control over the personalisation of the AI, being able to make it sound as they please. As well, they will be able to make the AI talk back to them, having a variety of voices to choose from. Also, it will be able to perform actions, like playing relaxing music or suggesting better environment conditions.

When it comes to sentiment analysis, it will be able to analyse the voice tone of the person, to provide a more comprehensive understanding of the user's emotional state. As well, it will have a voice activation feature for a spoken phrase like 'I need help'. It will have routine notifications for reminders to eat, sleep, drink water or take breaks.

When it comes to the individual's personal needs related to the mental health issues they struggle with, they will be able to add what to focus on. This way, it will also be useful for people that struggle with insomnia or have constant headaches. The app will know what colour to give them in order to help them improve their state.

7.2.2 HARDWARE UPDATES

The project currently uses just a LED module, but in the future, it will be customisable, from RGB LED strips to light rings or smart light bulbs. This upgrade will be suitable not only for a customer having a better ambient environment, but also for more massive situations such as concert light shows and night parties.

One other aspect that will be taken into consideration is power management. It will have more efficient solutions for power consumption to extend the operational life of the hardware components and save energy in the long run.

At the same time, it will have more automatic environmental control, aiming to be integrated with smart appliances that control temperature, humidity or other aspects. Like that, the user will just tell the chat how they feel, and the chat will know what to change in order for them to feel better. This will be useful for people that have mobility issues.

Home assistant compatibility

7.3 FINAL THOUGHTS

This thesis demonstrates the potential of integrating technology with mental health support. The Smart Ambient Lighting System provides a concrete example of how advancements in AI, mobile app development and IoT can create supportive environments tailored to individual needs. By addressing the areas identified in the 'Future Work' subchapter, the system can be further refined and expanded, offering even greater benefits to users.

The ongoing development and enhancement of this project highlights the importance of continued innovation in technology to address the growing mental health challenges of the modern age. Taking advantage of these advancements, more tools can be created that not only support but also significantly improve the well-being of individuals in various aspects of their lives.

8. BIBLIOGRAPHY

[1] R. L. Parsons-Smith, “In the mood: online mood profiling, mood response clusters, and mood-performance relationships in high-risk vocations,” 2015, Accessed: Jun. 16, 2024. [Online]. Available: https://research.usq.edu.au/download/b2ebdddc12ad26731a9aa2e64abf6d56409f52a571b70a0e451a4aadf9ec9bd3/7932684/Parsons-Smith_2015_whole%20%28updated%20version%29.pdf

[2] A. Hedge, “Linking Environmental Conditions to Productivity.” Ithaca, NY, 2004. Accessed: Jun. 16, 2024. [Online]. Available: https://ergo.human.cornell.edu/Conferences/EECE_IEQ%20and%20Productivity_ABBR.pdf

[3] A. J. Elliot, M. D. Fairchild, and A. Franklin, *Handbook of Color Psychology*. Cambridge: Cambridge University Press, 2015. doi: DOI: 10.1017/CBO9781107337930.

[4] T. H. Zhang *et al.*, “Increased anxiety and stress-related visits to the Shanghai psychiatric emergency department during the COVID-19 pandemic in 2020 compared to 2018–2019,” *Front Psychiatry*, vol. 14, 2023, doi: 10.3389/fpsyt.2023.1146277.

[5] OpenAI, “OpenAI.” Accessed: Jun. 07, 2024. [Online]. Available: https://openai.com/

[6] F. Å. Nielsen, “A new ANEW: evaluation of a word list for sentiment analysis in microblogs,” in *Proceedings of the ESWC2011 Workshop on “Making Sense of Microposts”: Big things come in small packages*, M. Rowe, M. Stankovic, A.-S. Dadzie, and M. Hardey, Eds., in CEUR Workshop Proceedings, vol. 718. May 2011, pp. 93–98. [Online]. Available: http://ceur-ws.org/Vol-718/paper_16.pdf

[7] L. Daskalska, S. Tarima, J. Meurer, and S. Young, “Child Anxiety and Depression During the COVID-19 Pandemic and Unmet Mental Health Care Needs,” *Child Psychiatry Hum Dev*, 2024, doi: 10.1007/s10578-024-01668-4.

[8] C. Van de Velde, “Youth loneliness in pandemic times: a qualitative study in Quebec and Ontario,” *Int J Adolesc Youth*, vol. 28, no. 1, 2023, doi: 10.1080/02673843.2023.2223671.

[9] K. Imran *et al.*, “Typical Mental Disorders among Teens,” *Pakistan Journal of Medical and Health Sciences*, vol. 17, no. 3, p. 1, 2023, doi: 10.53350/pjmhs20231731.

[10] Regina Maria, “STUDIU REGINA MARIA: Românii nu merg la psiholog, desi au nevoie de ajutor. 1 din 2 români se confruntă cu o problemă de natură psihoemoțională în 2022,” Oct. 2022, [Online]. Available: https://www.reginamaria.ro/articole-medicale/studiu-regina-maria-romanii-nu-merg-la-psiholog-desi-au-nevoie-de-ajutor-1-din-2

[11] M. E. Sharp, “10 Reasons Why People Refuse to Talk to Therapists.” 2017. [Online]. Available: https://www.psychologytoday.com/intl/blog/i-hear-you/201710/10-reasons-why-people-refuse-talk-therapists

[12] C. AI, “Character AI.” Accessed: Jun. 07, 2024. [Online]. Available: https://character.ai/

[13] D. Morena *et al.*, “Assessing the Impact of Conversational Artificial Intelligence in the Treatment of Stress and Anxiety in Aging Adults: Randomized Controlled Trial,” *JMIR Ment Health*, vol. 9, no. 9, pp. e38067–e38067, 2022, doi: 10.2196/38067.

[14] M. Bundo *et al.*, “How ambient temperature affects mood: an ecological momentary assessment study in Switzerland,” *Environmental Health*, vol. 22, no. 52, 2023, doi: 10.1186/s12940-023-01003-9.

[15] I. N. Fathina, D. M. Koerniawan, R. Kawakami, H. Hasebe, and R. Budiarto, “Influence of Humidity on Thermal Comfort for Worker Productivity in Hot-Humid Climate,” in *2019 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC)*, 2019, pp. 127–133. doi: 10.1109/ICSGSC.2019.000-7.

[16] R. I. Ahmad, M. Y. M. Yusof, K. Sopian, A. R. M. Rebi, Z. K. M. Makhbul, and N. K. Makhtar, “Optimizing Humidity Level and Illuminance to Enhance Worker Performance in an Automotive Industry,” *Journal of Applied Sciences*, vol. 10, no. 14, pp. 1389–1396, 2010, doi: 10.3923/JAS.2010.1389.1396.

[17] R. Küller, S. Ballal, T. Laike, B. Mikellides, and G. Tonello, “The impact of light and colour on psychological mood: a cross-cultural study of indoor work environments,” *Ergonomics*, vol. 49, no. 14, pp. 1496–1507, 2006, doi: 10.1080/00140130600858142.

[18] D. Jonauskaite, B. Althaus, N. Dael, E. Dan-Glauser, and C. Mohr, “What color do you feel? Color choices are driven by mood,” *Color Res Appl*, 2018, doi: 10.1002/col.22327.

[19] S. Pourroostaei Ardakani, X. Liu, and H. Xie, “A Data-Driven Study to Highlight the Correlations Between Ambient Factors and Emotion,” in *Computer Science and Engineering in Health Services (COMPSE 2021)*, Springer, 2021, pp. 109–128. doi: 10.1007/978-3-030-90737-0_9.

[20] J. P. Grodniewicz and M. Hohol, “Waiting for a digital therapist: three challenges on the path to psychotherapy delivered by artificial intelligence,” *Front Psychiatry*, vol. 14, 2023, doi: 10.3389/fpsyt.2023.1190084.

[21] S. D’Alfonso, “AI in mental health,” *Curr Opin Psychol*, vol. 36, pp. 112–117, 2020.

[22] K. Huckvale, J. Nicholas, J. Torous, and M. E. Larsen, “Smartphone apps for the treatment of mental health conditions: status and considerations,” *Curr Opin Psychol*, vol. 36, pp. 65–70, 2020.

[23] J. Torous, J. Nicholas, M. E. Larsen, J. Firth, and H. Christensen, “Clinical review of user engagement with mental health smartphone apps: evidence, theory and improvements,” *Curr Opin Psychol*, vol. 36, pp. 65–70, 2020.

[24] StatCounter Global Stats, “Mobile Operating System Market Share Worldwide.” 2024. Accessed: Jun. 19, 2024. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide

[25] Philips Hue, “Philips-Hue.” Accessed: Jun. 07, 2024. [Online]. Available: https://www.philips-hue.com/ro-ro

[26] Google, “Google Home.” Accessed: Jun. 07, 2024. [Online]. Available: https://home.google.com/welcome/

[27] Amazon, “Amazon Alexa.” Accessed: Jun. 07, 2024. [Online]. Available: https://alexa.amazon.com/

[28] LEDVANCE, “Smart Lighting Solutions.” 2024. Accessed: Jun. 20, 2024. [Online]. Available: https://www.ledvance.com/consumer/smart

[29] Tapo, “Tapo Smart Home Products.” 2024. Accessed: Jun. 20, 2024. [Online]. Available: https://www.tapo.com/en/

[30] A. Developers, “Android Studio.” Accessed: Jun. 07, 2024. [Online]. Available: https://developer.android.com/studio

[31] Kotlin, “Kotlin Programming Language.” Accessed: Jun. 07, 2024. [Online]. Available: https://kotlinlang.org/

[32] A. Developers, “Guide to App Architecture.” Accessed: Jun. 07, 2024. [Online]. Available: https://developer.android.com/topic/architecture

[33] OpenAI, “OpenAI Text Generation Guide.” Accessed: Jun. 07, 2024. [Online]. Available: https://platform.openai.com/docs/guides/text-generation

[34] E. Hub, “ESP32 Pinout Reference: Which GPIO Pins Should You Use?” Accessed: Jun. 07, 2024. [Online]. Available: https://www.electronicshub.org/esp32-pinout/

[35] Components101, “DHT11 Sensor Pinout, Features, Equivalents & Datasheet.” Accessed: Jun. 07, 2024. [Online]. Available: https://components101.com/sensors/dht11-temperature-sensor

[36] Components101, “HC-SR501 PIR Sensor.” Accessed: Jun. 07, 2024. [Online]. Available: https://components101.com/sensors/hc-sr501-pir-sensor

[37] Components101, “BH1750 Ambient Light Sensor.” Accessed: Jun. 07, 2024. [Online]. Available: https://components101.com/sensors/bh1750-ambient-light-sensor

[38] Components101, “RGB LED Pinout, Configuration, & Datasheet.” Accessed: Jun. 07, 2024. [Online]. Available: https://components101.com/diodes/rgb-led-pinout-configuration-circuit-datasheet

[39] G. Cloud, “Google Cloud.” Accessed: Jun. 07, 2024. [Online]. Available: https://cloud.google.com/

[40] Node.js, “Node.js.” Accessed: Jun. 07, 2024. [Online]. Available: https://nodejs.org/en

[41] Firebase, “Firebase Documentation.” Accessed: Jun. 07, 2024. [Online]. Available: https://firebase.google.com/docs

[42] GitHub, “GitHub Documentation.” Accessed: Jun. 07, 2024. [Online]. Available: https://docs.github.com/en/get-started

[43] HobbyKing, “Keyes RGB LED Module for Arduino.” Accessed: Jun. 07, 2024. [Online]. Available: https://hobbyking.com/en_us/keyes-rgb-led-module-for-arduino.html

STATEMENT REGARDING THE AUTHENTICITY OF THE THESIS PAPER *

I, the undersigned POPA SORINA-ANDREEA,

Identifying myself with CI series TZ no. 514269, CNP (personal numerical code) 6010428350023 author of the thesis paper MOODUP – A SMART AMBIENT LIGHTING SYSTEM USING SENTIMENT ANALYSIS CONTROLLED BY A MOBILE CHATTING APPLICATION

Developed with the purpose of participating in the graduation examination completing the educational level of BACHELOR'S DEGREE organized by the Faculty of AUTOMATION AND COMPUTING within the Politehnica University of Timişoara, session JUNE of the academic year 2023-2024, coordonated by CONF. DR. ING RĂZVAN BOGDAN, considering Article 34 of the *Regulation on the organization and conduct of bachelor/diploma and dissertation examinations*, approved by Senate Decision no. 109/14.05.2020, and knowing that in the event of subsequent finding of false statements, I will bear the administrative sanction provided by Art. 146 of Law no. 1/2011 – law of national education, namely the cancellation of the diploma of studies, I declare on my own responsibility that:

- This paper is the result of my own intellectual endeavour,
- He paper does not contain texts, data or graphic elements taken from other papers or from other sources without such authors or sources being quoted, including when the source is another paper / other works of my own;
- bibliographic sources have been used in compliance with Romanian legislation and international copyright conventions.
- this paper has not been publicly presented, published or presented before another the bachelor/diploma/dissertation examination committee.
- In the development of the paper I have used instruments specific for artificial intelligence (AI), namely gpt-3.5-turbo (name) open AI (source), which I have quoted within the paper / ~~I have not used~~ instruments specific for artificial intelligence (AI)[1].

I declare that I agree that the paper should be verified by any legal means in order to confirm its originality, and I consent to the introduction of its content in a database for this purpose.

Timişoara,
Date 23.06.2024

Signature

* The statement will be filled-in by the student, will be signed by hand by them and inserted at the end of the thesis paper, as a part of it.

[1] One of the variants will be selected and inserted in the statement: 1 – AI has been used, and the source will be mentioned, 2 – AI has not been used